

In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission from me.

7/25/68

THE PATH-GENERATION BY A PLANE FOUR-BAR CHAIN

A THESIS

Presented to

The Faculty of the Graduate Division

by

August J. Nechi

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Mechanical Engineering

Georgia Institute of Technology

October, 1968

THE PATH-GENERATION BY A PLANE FOUR-BAR CHAIN

Approved: _____

✓ ✓
Date approved by Chairman: 19-Dec-'68

ACKNOWLEDGMENTS

It is the author's desire to acknowledge the continued encouragements which came from the former department head, Dr. Kenneth Picha, who was instrumental for the start of his doctoral program and who took a sincere interest and concern in all particulars. It is also necessary to attribute a large portion of the successful investigation to the generous supply of institutional facilities, such as a large selection of native and foreign periodicals dating back to their first issues and an ample resource of analog and digital computer time. The author wishes to thank all the computer center advisors and especially Dr. Lawrence Gallaher for his expert advice in many problems in computer programing.

A major contribution was made by Dr. Frank Crossley, the thesis advisor, who suggested and motivated the problem with its possible attack of solution and who was able to secure a National Science Foundation grant which contributed some part to the financial support.

It is also necessary for the author to admit that he was a full fledged foe of any computer work and that it was the merit of Professor Robert Ashworth to make a first attempt of introducing him into the secrets of computer programing.

And finally but not lastly, the author must express a deep appreciation for his wife, Maria, who allowed him to resign from a well paying teaching position for the start of his doctoral research and who went to work by herself to be able to contribute a large portion of his support, which should be called the "wife's grant."

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS.	ii
LIST OF TABLES	vi
LIST OF ILLUSTRATIONS.	vii
SUMMARY.	ix
Chapter	
I. HISTORICAL INTRODUCTION	1
II. FOUR-BAR SYNTHESIS.	9
General Comments.	9
Coupler Curve	14
Computer Application.	16
Optimization Process.	18
III. ACCURACY POINT METHOD	22
General Comments.	22
Modified Grashof Theorem.	23
Four Accuracy Point Method.	32
IV. ANALOG COMPUTER USE	34
General Comments.	34
Specification of the Desired Curve.	36
Error Definition.	37
Analog Simulation	51

TABLE OF CONTENTS (Continued)

	Page
V. MATRIX INVERSION.	61
General Comments.	61
Matrix Inversion.	62
A. One-loop Method for the Solution to a Set of Inhomogeneous Equations.	63
B. Matrix Inversion.	68
C. Recognition of Independent Section and Rank of the Matrix	76
Matrix Inversion by the Least Square Method	77
The Pseudoinverse Method with Exact Solutions for a Selected Set of Equations	86
VI. DIFFERENTIATION METHOD.	89
General Comments.	89
Standard Estimate of Differentiation Accuracy	91
Circle Point Method	94
A. Derivative of the Approximation Circle.	95
B. Selection of the Finite Increment for the Desired Accuracy.	99
VII. GRADIENT METHOD	102
General Comments.	102
Hyperspace of Four-Bar Mechanism.	104
Definition of the Ninefold Optimization Vector $\bar{\Delta h}$	111
Definition of Stepsize by the Use of a Modified Regula Falsi Method	114
VIII. RELAXATION METHOD	118
General Comments.	118
Introduction to Relaxation.	119

TABLE OF CONTENTS (Concluded)

	Page
Relaxation for Computer Use	123
IX. JOUKOWSKI PROFILE	127
General Comments.	127
A. Base Line.	130
B. Core Line.	130
C. Airfoil Profile.	133
Actual Profile Coordinates.	135
Profile Synthesis	136
X. COMPUTER PROGRAM.	144
General Routines.	144
Four Accuracy Point Method.	147
General Matrix Inversion.	158
Analog Computer Simulation.	167
Computation of Gradient and Relaxation.	173
Main Program.	177
XI. CONCLUSIONS	180
BIBLIOGRAPHY	182
Literature Cited.	182
Other References.	189
VITA	192

LIST OF TABLES

Table		Page
1.	Changes of Link Dimensions for Nine Precision Points.	139
2.	Coordinates and Approximation Error of the Precision Points.	139
3.	Link Dimensions for Thirty Precision Points	140
4.	Coordinates and Errors for Thirty Precision Points.	140

LIST OF ILLUSTRATIONS

Figure		Page
1.	Semantics of Four-Bar Mechanism.	3
2.	Maximum of Nine Accuracy Points.	12
3-7.	Motion Limitations	25
8.	Integration of Error	39
9.	Summation Substitution for Error Integral.	39
10.	Error Definition Depending on a Parameter.	41
11.	Error Definition with Intersection Points.	41
12.	Notation Used in Analog Simulation	44
13,14.	Dwell Positions for Link z_2	48
15,16.	Inversion of Angle Differences	50
17.	Line Normal to Coupler Curve	50
18.	"Regula Falsi" Method.	54
19.	Approaching Difficulties of the Point C to the Position C^*	54
20.	Rotation of Coordinates about the Angle z_9	58
21.	Various Differentiation Methods.	93
22.	Approximation Circle	96
23.	Finite Increment Selection	96
24.	Z-Components of Nine Dimensional Hyperspace.	105
25.	Analysis of Gradient Vector.	109
26.	Modified "Regula Falsi" Method	116
27.	Graphical Representation of Transformation $\zeta = f(z)$	129

LIST OF ILLUSTRATIONS (Concluded)

Figure		Page
28.	Joukowski Profile.	131
29.	Preliminary Synthesis of Joukowski Profile	137
30.	Airfoil Path Generation, Initial and Final Stages	138
31.	Thirty Precision Point Synthesis of Joukowski Profile.	142
32.	Vectors in Complex Number Notation	153
33.	Compatibility Configuration.	153

SUMMARY

It was the objective of this investigation to show the implementation of an analog and digital computer combination for the curve synthesis by a plane four-bar mechanism. The available equipment made it necessary to simulate the analog computer on the digital machine for obtaining complete control of the analog operations by the digital component.

This analog simulation was adapted for the coupler point motion and is able to reproduce the complete path obtained from any plane four-bar chain including a double rocker chain. The key to the entire program is the idea of measuring the error between the obtained and the desired point positions around the path during several trial runs and of effectively minimizing these errors by an evaluation of the preceding trial runs.

A new and more accurate definition of the error between the achieved curve and the specified precision point is developed, which measured the length of the normal between the point and the curve. This, together with the large number of specified desired accuracy points and the introduced differentiation method, allows the process to be rid of any complicated analytical computations.

The developed method results in a powerful optimization process which consists of a combined relaxation and gradient mode. Relaxation is used for the initial steps in preference to the gradient method; the gradient method, however, is most effective for the final iterations.

An attempt is made to explain the observed phenomenon by the help of a multidimensional surface which, in general, must have a concave form with three equal minimums. Due to the multitude of dimensions and the high analytical order, it seems to have a great number of crevices which tend to block any orthodox minimization process.

The developed procedures are applied to a Joukowski airfoil as an example: 30 accuracy points are selected for fitting. However, it uses only nine points for its initial iteration steps because this uses less computer time and also demonstrates a better convergency. Once the roughing-in process is completed, the accuracy or precision point number is increased to 30.

The start of the program required the dimensions of a four-bar chain with a coupler curve which at least will surround the desired curve. If no such mechanism is known, the use of Sandor's four accuracy point method with some minor modifications is suggested because it serves the same purpose and uses less computer time than his five accuracy point method.

CHAPTER I

HISTORICAL INTRODUCTION

The attempt to synthesize a general plane curve by the utilization of mechanisms is not very old. In general a mechanism can become quite an elaborate and sophisticated piece of machinery and may have a rather long chain of various links which can become difficult to analyze for their displacements, velocities, and accelerations (jerks). Further, it is found that, in general, it is more economical to handle a given problem with the least possible number of links, because of the increasing manufacturing costs with a larger number of mechanism links. This leads to the crank-slider mechanism and the four-bar mechanism, of which the latter deserves special attention, because of the technological advantages of its link connections. These circumstances made it particularly interesting to discover how well a given problem could be solved by the application of such a four-bar chain and, therefore, the material selected for this presentation is the synthesis of a plane curve or the path generation by the use of a four-bar chain.

If, however, the generation of a straight line is considered as a special case of a plane curve, then one must admit that such attempts do go back as early as 1784. This was the time when James Watts used a four-bar mechanism for the approximation of a straight line motion for his reciprocating steam engine. It was needed to substitute for a plane surface of a slider which could not be machined at that time. The planer

was not invented until 1817. It was much later, in 1864, when a higher order device of more than four links was found by Peaucellier which was used for drawing a perfect straight line [1,2,3]. His problem setting was completely different from J. Watts'. In a few words, he was asking, how straight is a straight line? There was no question about a circle to be circular, if drawn by the help of a compass, but a straight line drawn by the aid of a ruler depended entirely on how straight the ruler was. He saw an imperfection in this method and invented a compass for straight lines, as one has others to draw ellipses and other plane curves.

There were many other approximations of straight lines. Always, the aim was to achieve the best results by the use of as simple a device as possible. A more recent dissertation by D. Tesar [4] indicates that there still exists an interest in the improvement of the straight-line motion generators. Attention turned to the general type of path generation about the year 1920; a graphical method to synthesize a curve through as many as five precision points [5,6] was developed. But it would not be fair to overlook all the spade work which was done up to this time. There is a vast amount of material of which only a few items have been mentioned.

First, there was Cayley and Samuel Roberts who published around 1870 an analytical expression for the coupler curve; that is the curve traced by a point attached to the link which has no pin connections with the stationary link (Figure 1) and is used in the synthesis of a path generator. The analytical form shown in most texts uses cartesian coordinates which have the x-axis in the direction of the frame $\overline{B_0A_0}$ and the

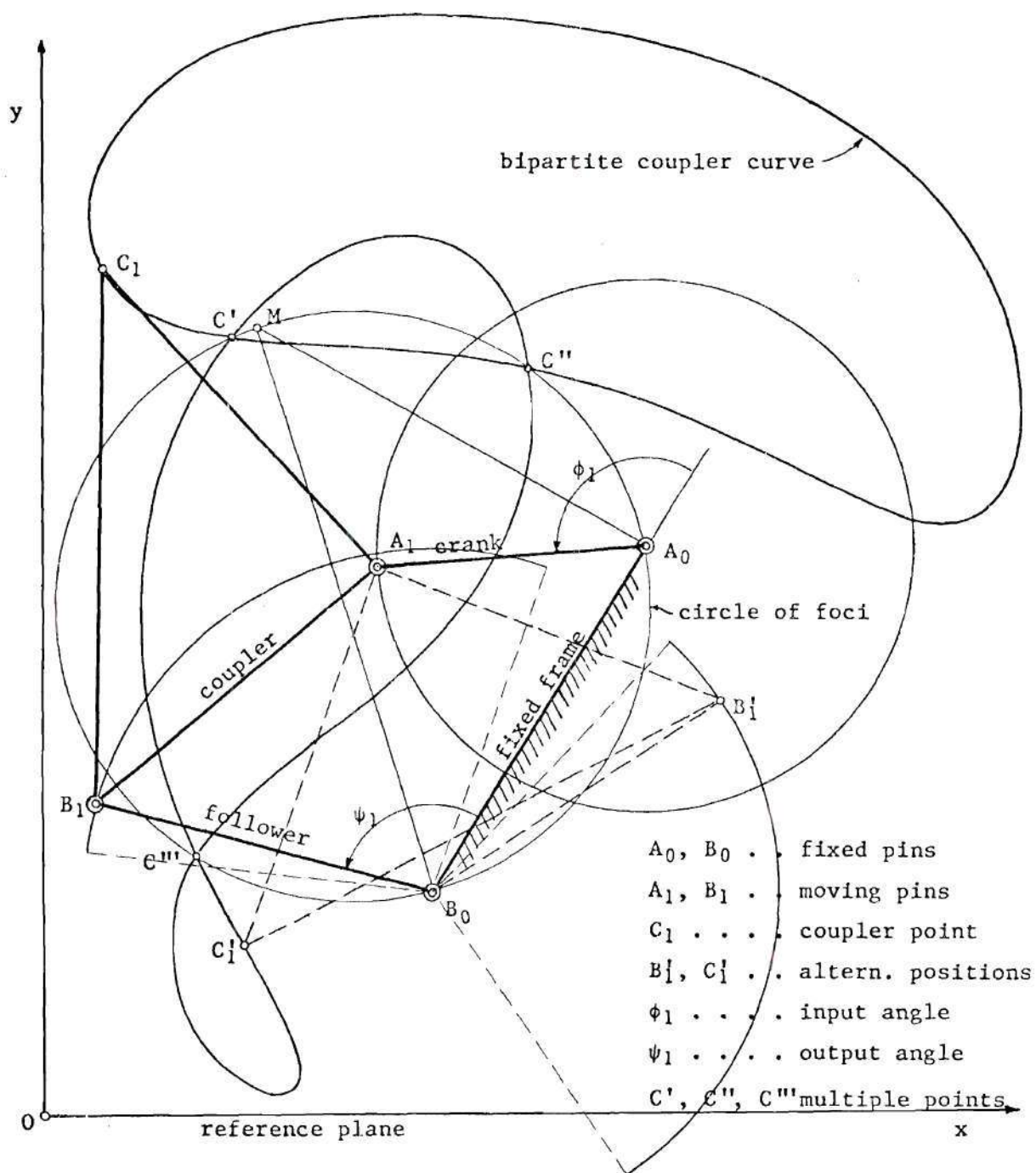


Figure 1. Semantics of Four-Bar Mechanism

origin in the point B_0 . This simplifies the expression considerably but was still general enough for the analysis made by the two mathematicians. It has a tricircular character and is of the sixth degree. S. Roberts discovered the existence of the three cognate mechanisms by purely mathematical deductions. That means that for each four-bar chain there exist two others which will be capable of generating exactly the same coupler curve [7,8,9,10].

The publications by Reuleaux, Burmester, Grübler and Müller do show a keen interest in fundamental research in the areas of pure analytic geometry and kinematics [11,12,13,14,16,17,18]. They appeared at the end of the nineteenth century and deal mostly with the general curve characteristics and the motion of a moving system through a series of intermittent positions. It was Müller who described in 1889 the multiple points of the coupler curve located on the circle of foci [15]. It is also interesting to see some early attempts to use elliptic functions for the coupler curve theory by Darboux [19] and the use of complex numbers by Morley [20]. In later periods the elliptic functions became almost forgotten and only the complex numbers were used at an increased rate [21,22,23,24]. Also, an attempt by Bennett to introduce a new coordinate system was not too successful [25]. He used the three focus points as coordinate reference.

There are numerous publications which all have some relation to the kinematics of the four-bar mechanism; some are collected in texts and others appeared in single papers. Beyer wrote a text with the title "Kinematische Getriebesynthese" which includes a multitude of original methods on the construction of coupler curves with special characteris-

tics [27] and it is interesting to read in his earlier text, "Technische Kinematik," on page 315 about the difficulties of solving the coupler curve equation for its nine parameters [26]. This would not be the equation as found by Cayley and Roberts but a more general form which uses a coordinate system as shown in Figure 1. This difficulty led to the idea of replacing this equation by an approximating expression which could be put into its explicit form and had less operational difficulties. These are the use of the Chebichev's polynomials [28] suggested by Bloch [29] and an interesting approach of using periodic functions by Meyer zur Capellen [30,31,32,33].

First attempts to generate a coupler curve through an arbitrary set of points are shown by Alt [5]. He found a graphical method which uses an inversion of the crank and follower motions to determine the pin positions A_1 and B_1 . This method will not really synthesize an entire curve but is able to find a mechanism with a coupler curve going through five points spaced at finite intervals. These points are called accuracy points or precision points. The obtained results are not unique and depend entirely on some random selections of items needed for the construction. This situation does indicate that it must be possible to increase the number of points, trading them against fewer random selections. Following this trend, a point reduction method was developed by Hain [34] which would allow an unlimited number of points; however, they cannot be of any general location. With this restriction on the desired precision points, the method lost its generality and became of limited use in its application. Bloch suggested a graphical comparison method [35] and the text by Hirschhorn shows a good collection of avail-

able graphical methods [36].

An analytic method which was adapted for the use of a digital computer was shown in 1959 by Sandor and Freudenstein [37,38]. However, it had the same restrictions as the graphical procedures with certain needed random selections and an upper limit of five precision points. The method uses complex numbers and does show very useful approaches to the computational problems involved. It is unfortunate that these selections for the free variables do not warrant the proper sequence of the precision points and represent a disadvantage against the graphical approach which allows an observation of the developing result during its construction. The resulting error of the curve sections between the selected precision points can become very large and depends entirely on the selections made for the free variables. A later paper by Freudenstein [39] suggests a spacing of the precision points to get better results and does show a function generator problem but no application to the path generator. The idea is basically the same as shown by Bloch who applied it to the straight line mechanism.

For the path generator with a large number of precision points, it became difficult to define the obtained error, and various methods have been suggested [39,40]. They are more or less approximations only which, in some cases, may yield misleading results. There are more papers published for function generators with a larger number of points [41,42] than there are for path generators [43,44,45], but almost any path generator method could also, with some modifications, be applied to the function generator problems. In general, it is a tiring experience to find in most of these publications an insufficient amount of

information which makes a detailed check on the claimed results impossible. It seems that some of the published procedures and computer programs may not converge to the desired end result as well as they are demonstrated for function generators and need to be read with caution. In general, it should be noticed that certain procedures which are operating for function generators may very well not be operating for path generators where the conditions are much more severe. A function generator represents a mechanism with an established relation between the input and the output angles. This eliminates the positioning of the frame link in relation to the reference plane and the positioning of the coupler point C in relation to the moving plane of the coupler link. Also the error definition is rather simple compared to the problem of finding the perpendicular distance of a point to the coupler curve (see the section on error definition in chapter IV).

With the indefinitely increased number of desired precision points the synthesis procedure seems to call for a method with a continuous output facility for the coupler point coordinates. Such devices have been found in the analog computer which is capable of tracing coupler curves on coordinate plotters or on a scope with two input circuits [46,47,48,49,50]. Again, special caution is needed because some proposed methods will work only for well behaved situations and cannot be used in general for any type of four-bar mechanisms. None of the published methods can be applied to mechanisms which have an input link with a rocker motion. Such an input link cannot rotate through 360 degrees but does reverse its motion and would require an extra logic circuit for this purpose. All these methods are used for the curve generation of a given mechanism

and are not used for any synthesis problems.

In our computer age with facilities working with the speed of light, one would desire to make best use of these available tools and one should proffer for the path generator problem the facilities of a hybrid computer which can combine the functions of a digital and an analog computer. The presented investigation does show how it is possible to make such a use of the combination of these two computer systems for the synthesis of a path generator.

CHAPTER II

FOUR-BAR SYNTHESIS

General Comments

The complexity of the attempted synthesis problem rests entirely in the structure of the analytical expression for the coupler curve of the four-bar mechanism. The path generator requires a process which will modify this coupler curve to take a shape and position as close as possible to the desired curve called the path. In the ideal case, this path and the coupler curve should become identical and only, if such a solution cannot be obtained, then the path and the coupler curve should be fitted as close as possible. With the varieties in the possible shapes of a coupler curve it becomes impossible to predict which desired path can be fitted perfectly and which one can only be approximated.

The subject matter of this investigation shall be to generate the path representing the locus of various point positions of a traveling point and should be well distinct from a different problem which deals with the positions of a traveling plane. In order to specify the positions of a traveling plane in a two dimensional space, one requires the knowledge of the motion of a second point which is moving equidistant to the first one. This, in fact, would require the simultaneous generation of a second path fulfilling the requirement of having certain specified equidistant points. Burmester published in his text [12] a graphical method for such a synthesis of five positions of the moving plane which mostly results in more than one solution but in rare cases may not yield

any results at all. More recently this method was computerized and helps to synthesize the motion of a plane through five positions [51]. If solutions exist, the coupler plane will move exactly through the specified positions but will not meet any other requirements between those five positions.

Also the methods for generating a single path do include graphical methods of up to five accuracy or precision points. The methods with a higher number of points lost their generality as it was mentioned on page five. Some of these accuracy point methods were computerized but do not exceed five precision points. Whatever advantage there may be in a free selection of the input angles, these methods cannot be considered satisfactory in a theoretical point of view. To explain this statement better, it might be best to compare it with the synthesis of a circle through three point positions. It is well known that three points in a plane are sufficient to specify a circle but also two points will specify a circle with two different positions if one has preselected a radius for it. That means one has traded one precision point against one random selection. I admit that the synthesized circle through the three positions may have an inconvenient radius but one should not deny the fact that it has a great value to know that three points will always synthesize uniquely a circle with a specific radius which is found by the synthesis. Coming back to the coupler curve problem, this would mean to say, how many precision points do specify uniquely a certain coupler curve which is derived from a specific four-bar mechanism. Once such a mechanism is found it is known that by the Roberts' Theorem [10] there exist two other four-bar mechanisms, the cognates, which will produce an

identical coupler curve. In this manner each solution actually represents a threefold solution.

As it was remarked by Beyer [26], it should be possible to define a coupler curve uniquely by specifying nine accuracy points (see also page 4). This would represent an increase in precision points from five to nine by trading the additional precision points against four input angles. (The starting position of the input crank is used as reference position.) Several test runs of the developed procedure verified this statement; however, it was found that no arbitrary sequence can be used for the selected nine points (see Figure 2). For the circle example, the sequence does not matter, because it is possible to follow any chosen point succession by maintaining a clockwise or counterclockwise direction. However, it is not possible to do the same with four or five points on a conic. Here the sequence is important and is usually found by inspection. The increased number of points to nine for a coupler curve, however, makes the selection of the proper sequence almost impossible due to the diversity of the coupler curve forms.

It is true that the analytical computation of a conic through five points normally does not need any point sequence and that the obtained result will represent the only solution. However, for the coupler curve, if specified by nine points, no such analytic method is available. If it were, it would most likely turn out some surprises in the point sequence as the one shown in Figure 2. It would be possible to specify the required four-bar mechanism, but it would not guarantee any sequence nor the fact that all nine points should be on one curve. The shown figure illustrates what is to be expected from an analytical accuracy

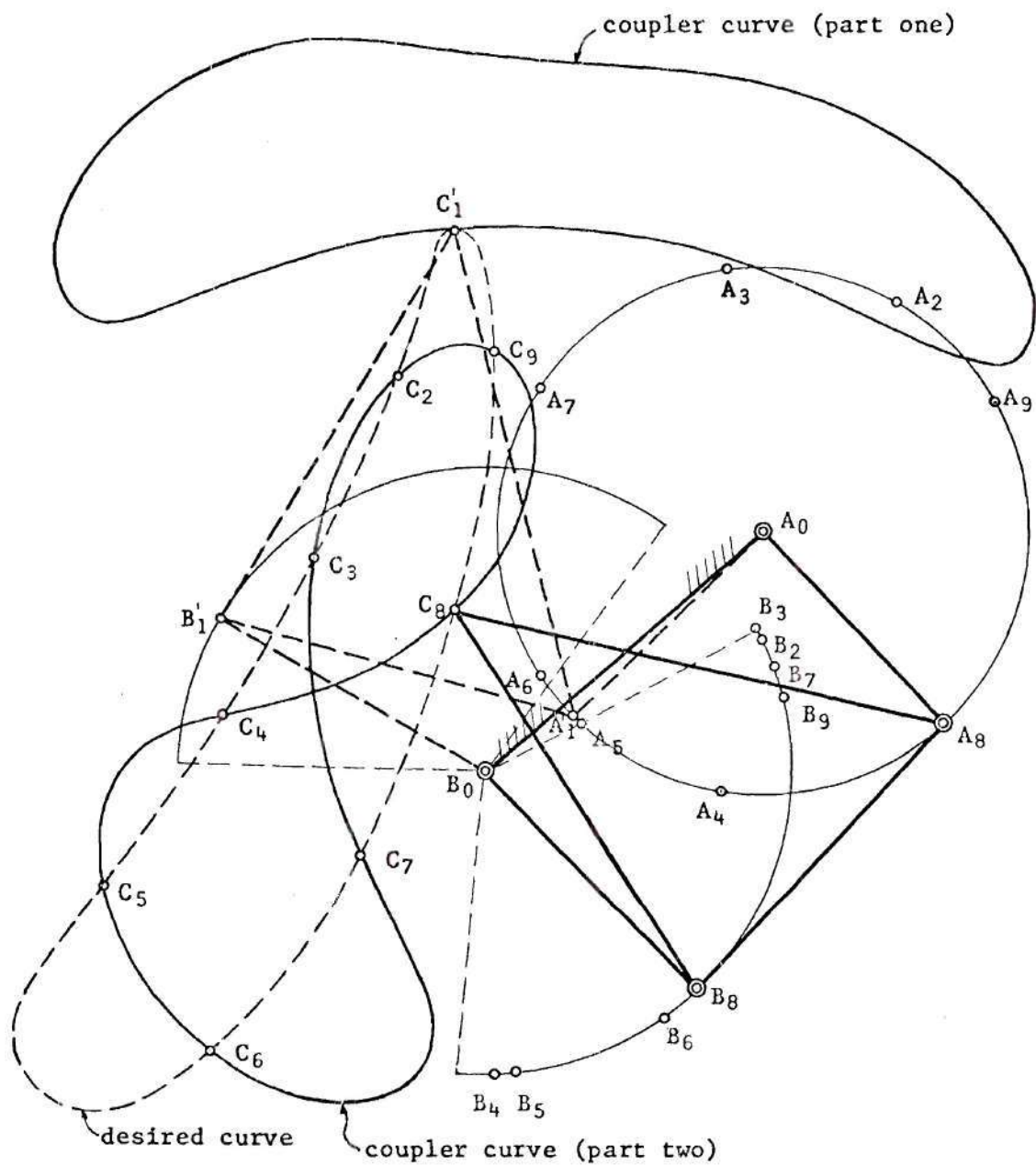


Figure 2. Maximum of Nine Accuracy Points

point method for nine points selected on a desired curve. Point one is on a different part of the coupler curve as all other eight points and the points from four to seven have a wrong succession. This type of result can already be found with the methods proposed by Sandor [37] for four and five accuracy points. In his methods, a correction can be made by changing the assumed angles and finally may, after several trials, render the desired coherent curve. But any fictitious nine accuracy point method would not allow an angle correction and the result shown would simply indicate that, if the nine points are selected in the wrong sequence, it is not possible to synthesize these points by a coherent curve.

The only remedy for this situation would be a relocation of the nine points on the desired curve. This may eventually render a result of a mechanism with a coupler curve through all nine points in the required order, but it will not be able to include the first set of points shown in Figure 2. It then becomes obvious that it should be possible to approximate the desired curve as shown, but the coupler point C will not have its path exactly through the nine selected points but will represent a close approximation of the desired curve. In this manner, it will become the task to find the best distribution of the nine accuracy points on the desired path. It should be noticed that the desired and the obtained path will always have an even number of intersections or accuracy points if the paths are given as closed loops.

Such a method would require a continuous repetition of a given accuracy point method with a comparison of previous results until a best solution is found. The quality of the result can be best determined by

taking measurements of the positioning error at certain crucial points of the desired path. This perception, together with the convenience to specify any shape of path by a larger set of points, helped to aim this investigation towards a multiprecision point method with the aim to minimize the sum of all obtained positioning errors.

Coupler Curve

The coupler curve formed by the path of the coupler point on a four-bar linkage system represents the core of this simulation technique. The variety in its form appears to make it possible to generate almost any type of closed or open curve. This is, if one will allow the mechanism to take any dimensions and if one excludes certain types of curves where the four-bar simulation seems to be hopeless, such as spirals, larger sections of involutes, cycloids, and trigonometric curves or, for example, multi-leaved roses. The emphasis is here on the generated path and not on the practicality of the mechanism, or in other words, on how close a given path can be synthesized without any restrictions to the physical dimensions of the generating four-bar linkage system.

If this curve can assume such a variety of forms, it becomes interesting to study it more closely and to find, if possible, some clues as to what shape it could assume. In general, it is a curve of sixth order. For example, the bipartite curves shown in Figures 1 and 2 are of sixth order if the two sections are counted as one unit, but it can be seen that each section alone forms a curve of the fourth order. It is found that the closed section could assume a second, a fourth, or a sixth order. The sixth order section will not have any second part, be-

cause the analytical expression for this curve shows an equation of sixth degree and the curve, therefore, cannot assume any higher order. However, if one section is of second order, the other one must be of fourth order to give a total of six, that is, it must be possible to place a straight line in such a way that it will have six intersections with the two curve sections.

The interest in the analytic analysis of higher order curves is not new. The mathematician Roberts [52,53] already analyzed the coupler curve and was able to describe some of its characteristics. It is called a tricircular sextic because it has three imaginary circular or cyclic points out in infinity. The designation for those points originates from the fact that any ordinary circle has also the same cyclic points out in infinity and two imaginary asymptotes. Similarly, there are two sets of three parallel asymptotes with the coupler curve. They are all imaginary lines but intersect at three real and six imaginary points, the so called singular foci [54]. The sides of the triangle between the three real foci form the base or frame links for the three cognate mechanisms which have a common coupler curve. The curve itself may have as many as three double points which can form cusps or crunodes and one section can be of second, fourth, or sixth order.

All this knowledge does help to make a preliminary decision whether or not a particular path has some prospects to be simulated by the four-bar mechanism but will not allow an actual generation of the path with a minimum deviation. There are graphical methods which show how to obtain certain curve characteristics or accuracy points, but they do not allow the simulation of an entire curve.

The main obstacle for the curve simulation is given by the sophisticated analytic equation for the coupler curve. There was no effort spared to simplify this equation to a more workable form by the use of complex numbers, elliptic functions, and the introduction of new coordinate systems. The observation shows further that due to the high non-linearity of sixth degree, any linearization attempt for the desired optimization fails to furnish the desired result. It is for this reason that a more powerful and more general applicable optimization method had to be found.

Computer Application

After a thorough study of the past and present methods for the path generator and in view of the difficult expression for the coupler curve, an attempt was made to avoid some of the obstacles by using a point by point approximation method. This, in turn, calls for a repeated operation which has to evaluate the diversions of the point locations and, therefore, becomes a typical computer application. For an increased accuracy of the generated curve, an increased number of precision points is needed and a more powerful and faster computer equipment becomes desirable. It was for this reason that it was felt it would be best to apply the facilities of a modern hybrid computer. The analog components of this type of computer with its speedy response could be used to its best advantage wherever possible, and the digital component would serve as an auxiliary to control initial conditions and to take readings of final and intermittent results. However, efforts to receive access to the required hybrid failed, and a compromise of an analog simulation on the digital computer had to be substituted. It is obvious that this de-

vice was not able to fully replace a hybrid computer but made it possible to analyze the prospects of such a computer combination.

Any such optimization process will be most successful if the starting conditions are not too far off the desired end result. This should not mean that the developed process will fail but that a good start does represent a time saving on the computer. The mentioned four accuracy point computer method should serve this purpose and was adapted to this application. It was selected to save some computer time because it is less elaborate compared with the five accuracy point method. However, it was found that both methods, using four or five points, cannot foresee whether or not all points will have the proper sequence and whether or not they all will be on the same curve, in case of a two sectional coupler curve. This malfunctioning of the initial approximation may render a rather poor starting mechanism which could be poorer than any other initial guess known from other sources [55]. For this reason, a final testing process was added, but it still is suggested that one should use this accuracy point method only for cases in which no other guess is available.

The four accuracy point method [37] requires five angle selections which are fed as input data to the computer. Three of them can be selected at random, but the remaining two must be within a certain range. The developed procedure will automatically assume more than one angle within the permissible range and finally test the resulting mechanisms for their physical proportions and a coherent curve for all four accuracy points. The mechanism with most uniform link lengths is selected and processed further to be optimized by the combined gradient and relaxation method.

It was not possible to find a simple check for the sequence of the desired point positions at that point of the procedure.

Next, the selected result is transferred to an analog type simulation program which takes actual readings on the diversions of the obtained point locations and will include all specified precision points in the desired sequence. Here, no limit is set to the number of points. The readings are recorded and added together to a sum of all absolute values of these errors which indicates the quality of the obtained result. The hybrid facilities would also permit making the curves visible on a scope screen and a more direct supervision of the computer progress could be achieved. The analog simulation was modified for the use with rocker inputs. Those are driver links which cannot rotate through 360 degrees. This change was obtained by the use of the digital components and a modified Grashof Theorem [56], explained in chapter III. The readings and the specified points will be taken step by step and in the sequence as they are supplied in the computer input data. In this form the minimization process will tend to correct possible contingent misconceptions in the sequence and the continuity of the curve.

Optimization Process

After all of this previous preliminary work, the minimization of the resultant error between the desired and the actual curve became the salient point of this investigation. In this respect, it was interesting to read a publication by a mathematician in Zürich, Professor Stiefel [57], which confirmed the suspicion that for some optimization problems none of the conventional methods seemed to work satisfactorily and that the so

called relaxation method appears to be more promising than any other. If one could apply some geometric analysis in its wide meaning to this problem, one should be able to explain better the reasons for this behavior.

It is known that the general coupler curve of a four-bar chain has nine parameters which are independent and do specify a certain mechanism including its position in the fixed reference plane. The independency could be assumed for a wide range and would have its limit at those boundaries for which the four link chain becomes unable to close to a moving linkage system. This situation occurs when one link becomes longer than the three remaining ones added together. However, if the desired coupler curve requires a mechanism with a wide sweeping motion, this boundary should not be approached because of the rather limited mobility in its close neighborhood. Any link motion will die out at this boundary and, if the optimization process is effective, one should not need to be concerned about its existence.

Considering this independency of the nine components, it must be possible to represent each mechanism by a nine dimensional vector in the nine dimensional space of those components. It is one step further to see the accumulated coupler curve error of each mechanism as an additional dimension added to the nine dimensional space. This dependency of the tenth dimension will render a surface in the ten dimensional space.

This formulation of the problem would be similar to the presentation of a surface in a three dimensional space formed by a z-component dependent to each point of the x-y plane. In the presented situation, the x-y plane would compare to a media with nine dimensions to which the

tenth dimension is added.

The character of such a surface does depend on the degree of its analytical equations. As one can see from the two dimensional curves, the number of depressions and their depth are increasing with the degree or the powerfactors of such equations. This applies even more so to a three dimensional or to a ten dimensional surface. In view of this consideration, it should be realized that the surface which represents the error function must be filled with a large number of deep depressions but has basically a concave form with three equal minimums, of which each represents a cognate mechanism of the other minimum.

Considering this representation, it becomes evident that a pure steepest descent or gradient method will most likely be trapped in the many crevices without being able to find one of the desired minimums. However, the relaxation method applies a different principle to the problem. It follows the general outlines of the surface taking into consideration only major trends in its form. However, the practical experience shows that also this approach can become deadlocked. In order to free the procedure from such a halt, the procedure will change back to the gradient method and stay with it until it cannot achieve any further progress. The procedure will alternate between these two basic methods and pull the computations from the deadlocked positions until the desired result is found. It must act in a similar way as the overrelaxation suggested by Southwell [58,59]. The demonstrated problem shows the capabilities of this optimization process. If this method is applied to other problems with different peculiarities, it should be realized that certain constant factors of the presented program may need to be changed

and are subject to the skill of the operator.

It should be noticed that the process has no built in restrictions for the purpose of limiting any velocity or acceleration factors. It was contended that it must be more important to find out whether or not a good approximation of a given curve is possible without hampering the process by any type of restrictions.

CHAPTER III

ACCURACY POINT METHOD

General Comments

The accuracy point method developed by Sandor and Freudenstein [37,38] accommodates as many as five accuracy points but requires a free selection of four angular displacements which are related to the second, third, fourth, and fifth accuracy point relative to the positions of the first one. The four accuracy point method is somewhat simpler in its computations but requires the choice of five angles which, again, relate the positions of the driver and the driven link to the chosen accuracy points. In general, the fewer the specified accuracy points are the more parameters have to be selected and their selection has a deciding effect on the usefulness of the end result. Or in other words, a larger number of desired accuracy points will require the selection of fewer parameters and will yield a more unique solution; and, as it was stated by Rudolf Beyer in his book Technische Kinematik on page 315 paragraph 189 [26], it should be possible to use as many as nine accuracy points. This would eliminate all selective parameters and should yield a unique solution without the need of selecting any parameters. However, as he further states, the character of the required nine equations, that is one for each point, is so complex and highly nonlinear that no solution was obtained so far.

The presented investigation does not limit itself to any number of accuracy points and uses the four accuracy point method only for the

initial approach to the path generator. It was noticed that both the selected four and also the five accuracy point method for the digital computer may not render the desired sequence or curve continuity due to the selected points with their related parameters. As explained in the General Comments to chapter II, the outcome of such an accuracy point method depends entirely on the selected angles for the crank positions, and even worse, because some coupler curves may have two independent sections, the procedure may produce a mechanism with a few points on one part and the remaining on the other part. Although the possible two sections are derived from the same mechanism, they cannot be reached by a complete turn of the mechanism. In order to recognize such solutions, a modified version of the Grashof Theorem was applied and in recognizing these difficulties only a four accuracy point method was selected and improved to serve as a preliminary approximation for the actual path generator. However, it is recognized that even a poor preliminary approximation may be just as good for the start of the path generator synthesis and that such a known preliminary guess would entirely eliminate the difficulty of selecting the proper crank angles for each accuracy point.

Modified Grashof Theorem

The Grashof Theorem [56] says, if the sum of the lengths of the longest and the shortest link is less than the sum of the lengths of the other two links, then the shortest link is able to rotate through a full revolution of 360 degrees. This makes it necessary to find the longest and shortest links before any mechanism can be tested for its mobility. It was found that in the presented procedure it is not impor-

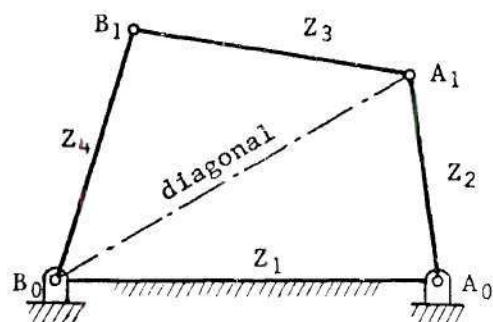
tant to know which of the links, if any, can rotate. Such an analysis is needed only for the driver and in rare cases for the follower. It, therefore, became more profitable to apply a modified version of this theorem.

Let Z_1 , Z_2 , Z_3 , and Z_4 define the lengths of the frame, the driver, the coupler, and the driven link or follower, as shown in Figure 3. Considering the Grashof's condition, there can be two stipulations; either they are or they are not satisfied. For the first case, if they are satisfied and when realizing that the role of the driver and the follower is interchangeable, then there are three basically different cases of motions to be defined.

1. The frame with the length Z_1 represents the smallest link. Here the three other links Z_2 , Z_3 , and Z_4 will be able to rotate through 360 degrees.

2. Either the driver or the follower represent the shortest link. Here this shortest link will rotate through a full revolution and the other ones will have only a limited rotation. In this way the shortest link will act as a crank and the other ones as rockers.

3. The coupler link does represent the shortest link. Now, neither the driver nor the driven link will be able to rotate through a full revolution. Only the coupler will rotate through 360 degrees and its coupler point will still describe two independent closed curves. It should not be too difficult to see that every one of these three cases has two independent closed curve sections which can only be obtained, one after the other, by disassembling the mechanism after completion of the first section and by reassembling it in its symmetry position about the



Z_1 . . . frame
 Z_2 . . . driver
 Z_3 . . . coupler
 Z_4 . . . follower
 A_0, B_0 . fixed pins
 A_1, B_1 . moving pins

Figure 3.

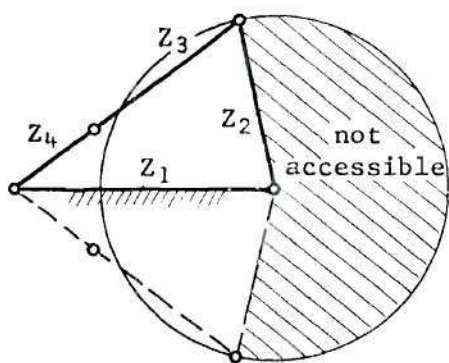


Figure 4.

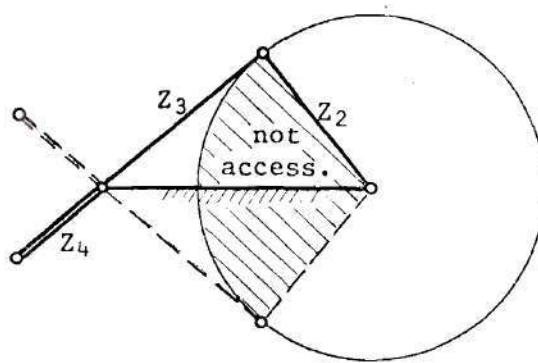


Figure 5.

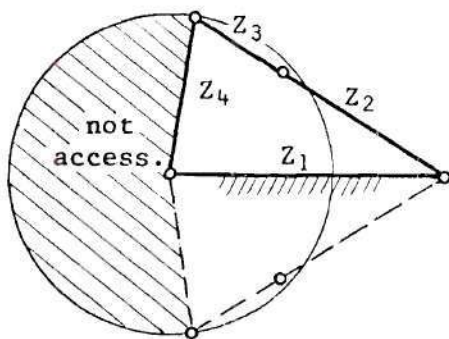


Figure 6.

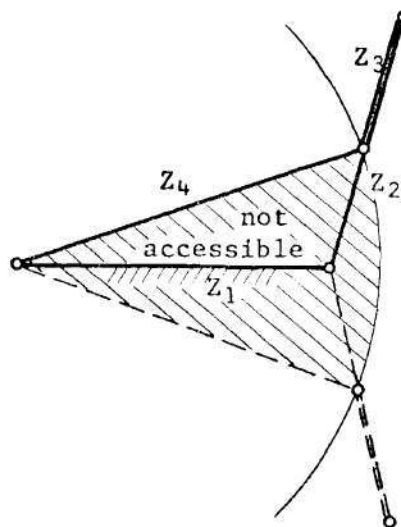


Figure 7.

Motion Limitations

frame link or about one of the diagonals in the quadrilateral. It is in no case possible to reach both of these sections without a reassembly job of the mechanism.

Cases with equal link lengths will rarely occur and are not included in this analysis. This omission can be justified because the computer almost never will develop two equal real numbers. Each real number carries a random error in its machine representation and will make it almost impossible to receive two links of identical length.

For the second condition, if the Grashof criterion is not satisfied, then none of the links can rotate through 360 degrees and the obtained coupler curve is a closed curve with only one section. Here it is not possible to assemble the chain in any other way if the sequence of the links from one to four is to be maintained.

These various cases may cause too many repeated operations in the computer program; therefore, it should be shown whether the evaluation of the two following inequalities could serve all requirements for the computations.

$$Z_1 + Z_2 \cong Z_3 + Z_4 \quad (1)$$

$$|Z_1 - Z_2| \cong |Z_3 - Z_4| \quad (2)$$

The Boolean expression with the inequality (1) will add the length of the frame to the length of the driver and compare it with the sum of the lengths of the coupler and the driven link. The Boolean of inequality (2) has the differences of the same quantities but taken as absolute values and with an inversed inequality sign.

The analysis does show that it is possible to state:

- a. If both relations are true of a particular four-bar mechanism, then link two can rotate through 360 degrees.
- b. If only one of the relations is true, then link two is restricted in its motion and none of the links can fully rotate.
- c. If both relations are false, then either link three or link four does freely rotate but link two has two separated areas in which it can move within their boundaries.

This leaves the question whether or not it is possible to derive from equations (1) and (2) a clue which will indicate the link, if it exists, which does rotate through a full rotation.

The above relations were used in Sandor's accuracy point method, but for the following proof it will be better to analyze the various possibilities from the inversed inequalities which will eliminate the equal signs.

$$Z_1 + Z_2 > Z_3 + Z_4 \quad (3)$$

$$|Z_1 - Z_2| < |Z_3 - Z_4| \quad (4)$$

Figure 4 shows that a true equation (3) will make it impossible for link two to rotate beyond the stretched position of the links three and four. They are forming one side of a triangle together with the remaining links one and two. There are two such positions limiting the motion of link two, either in a clockwise or in a counterclockwise direction, and they are symmetric about link one. Figure 5 shows the situation for the case of a true equation (4). Here the links three and four are folded and again form a triangle together with the two others. Link two cannot enter the indicated area. The absolute signs are needed to assure a valid result

for which Z_2 or Z_4 may be larger than the other values of the pair. Again a symmetry position about link one does exist. Hence, equation (3) indicates a restriction of the motion of link two in the upper and the lower area symmetric about link one and equation (4) indicates a similar restriction of link two in the opposite direction. This proves theorem one.

Theorem 1. Equations (3) and (4) indicate a restriction of the motion of link two and the type of restriction.

Corollary: If both equations (3) and (4) are true, the mechanism will not be able to reach its symmetry position without disassembly.

This can be easily understood, because link two cannot reach the symmetry position without going through any one of the inaccessible areas.

It remains to show which one of the links, if any, is able to rotate through a full rotation. This is done by omitting the absolute signs of relation 4 and by analyzing the resulting cases.

1. If $Z_1 > Z_2$ and $Z_3 > Z_4$ then the absolute sign can be omitted to the form

$$Z_1 - Z_2 < Z_3 - Z_4 \quad (5)$$

2. If $Z_2 > Z_1$ and $Z_3 > Z_4$ then the relation has to be set into the form

$$Z_2 - Z_1 < Z_3 - Z_4 \quad (6)$$

3. If $Z_1 > Z_2$ and $Z_4 > Z_3$ then one has

$$Z_1 - Z_2 < Z_4 - Z_3 \quad (7)$$

4. If $Z_2 > Z_1$ and $Z_4 > Z_3$ then it follows

$$Z_2 - Z_1 < Z_4 - Z_3 \quad (8)$$

Now equation (3) together with the parts of equation (5) will represent the following inequalities.

$$Z_1 > Z_2 \quad (5a) \quad Z_3 + Z_2 > Z_1 + Z_4 \quad (5b)$$

$$Z_3 > Z_4 \quad (5a) \quad Z_1 + Z_2 > Z_3 + Z_4 \quad (3)$$

Adding equations (5b) and (3) will supply the equation

$$Z_2 > Z_4 \quad (5c)$$

And further substituting the equations (5a) and (3) will develop

$$2 Z_1 > Z_1 + Z_2 > Z_3 + Z_4 > 2 Z_4 \quad \text{or} \quad Z_1 > Z_4 \quad (5d)$$

The new equations (5c) and (5d) together with the parts of (5a) allow the ordering of the links according to their lengths in three different ways and show that link four is always smallest and that either link three or link one is the longest one.

$$a. \quad Z_1 > Z_2 > Z_3 > Z_4$$

$$b. \quad Z_1 > Z_3 > Z_2 > Z_4$$

$$c. \quad Z_3 > Z_1 > Z_2 > Z_4$$

Equation (5b) indicates that the cases a and b do satisfy the Grashof requirement and that link four does rotate through 360 degrees. Equation (3) indicates that case c also satisfies the Grashof equation and, again, link four is able to rotate through a complete revolution. In an equal manner, it is possible to show that, in equations (6), it is also link four which can rotate through 360 degrees; however, for equations (7) and (8), it is link three which does rotate freely.

It is interesting to look into the last two cases to see what type of restriction link four may have, whether or not it acts like a rocker similar to link two for the two first cases, 1 and 2, with the combined

restrictions as shown in Figures 4 and 5. Analyzing the third case, the inequalities 7 can be rearranged to obtain

$$Z_1 + Z_4 > Z_2 + Z_4 > Z_1 + Z_3 > Z_2 + Z_3 \quad \text{or} \quad Z_1 + Z_4 > Z_2 + Z_3 \quad (7a)$$

Also the inequality 3 can be rearranged to the form

$$Z_4 - Z_1 < Z_2 - Z_3 \quad (7b)$$

Equation (7a) indicates a restricted motion for the link four as shown in Figure 6, and equation (7b) indicates a restriction for the same link in the other direction as illustrated in Figure 7. This would mean that link four acts like a rocker with restrictions in its motion in both directions. That is, provided one can show that equation (7b) has positive values on one side, to insure the proper direction of the inequality sign and no absolute signs with a similar meaning as in equation (4) would have to be added.

When equation (3) is combined with the conditions for the case 3, it is found that

$$2 Z_1 > Z_1 + Z_2 > Z_3 + Z_4 > 2 Z_3 \quad \text{or} \quad Z_1 > Z_3 \quad (7c)$$

And again it is possible to order the links in three different arrangements.

- a. $Z_1 > Z_2 > Z_4 > Z_3$
- b. $Z_1 > Z_4 > Z_2 > Z_3$
- c. $Z_4 > Z_1 > Z_2 > Z_3$

That indicates that Z_3 is the smallest link and that the right hand side of 7b must always remain positive. Hence, it is found that the inequality sign of equation (7b) is proper and the link four has two restrictions, one to each side. But from equations (3) and (7), one can also see that

link two has the same type of restrictions and, similarly, can only act as a rocker. Should, however, only one of the relations in the pair 3 and 4 be false, then none of the links has an unrestricted motion, because it is not possible to satisfy the Grashof's condition. There will be only one type of restriction in link Z_2 and in link Z_4 , depending on which relation became false.

Should both relations of the set 3 and 4 be false, then link two will trade its role with four and link one with three. The rest remains the same and one could again undertake the same analysis with four different cases when eliminating the absolute signs of equation (4).

This indicates that equations (3) and (4) establish the motion of the mechanism and it is possible to set up the following theorems.

Theorem 2. If the angular motion of link two is restricted as indicated by equations (3) and (4), then either the link four or the link three can rotate through an angle of 360 degrees.

Theorem 3. If both links two and four have a double restriction and do act as rockers, then it is the link three which can freely rotate through 360 degrees.

Theorem 4. If there is only a single restriction on link two, then none of the links can make a full rotation and every one has a single restriction only.

Corollary: Using the established facts of inversion, it becomes clear that, if link three rotates freely and is interchanged with the frame (link one), then all other three links will be able to rotate through a full rotation.

For the presented procedure, it was important to establish the

mobility of the driver which was sufficiently specified by both equations (3) and (4), and only in a very few cases the mobility of the driven link was needed which was then easily established by a modified set of relations similar to the ones for link two.

The computer evaluated these relations in a Boolean expression and established the restricting angles as needed. The entire process was combined in a procedure which can be called and modified as required by the main program.

Four Accuracy Point Method

The four and five accuracy point methods can be found in Sandor's dissertation [37]. Here the four accuracy point method is suggested with a few modifications.

Instead of selecting three displacements of the coupler angle, those of the input angle were chosen by a tentative guess. Should it happen that this choice yielded no useful solution at all, then a new selection for the same three angles was required and the halted program was to be repeated. The required additional displacement angles of the coupler and follower links must be selected within the compatibility limits. However, as it is known, these limits are bounding two separate angular areas and allow a multitude of solutions. Here, one solution out of each area was selected and, with all permissible symmetry positions rendered at least two or more solutions.

With these multiple solutions at hand, it is possible to build upon Sandor's work and to proceed to select the one with all accuracy points on one curve by the use of the modified Grashof Theorem. For this purpose, equations (3) and (4) are evaluated. Then the type of re-

strinctions obtained will make it possible to establish the motion pattern of links two and four. A check of the displacement angles against the limiting positions of the obtained mechanisms will allow recognition of all useful solutions.

CHAPTER IV

ANALOG COMPUTER USE

General Comments

As it was mentioned earlier, it was the intention to use the facilities of a hybrid computer for this analysis. When it became apparent that no such facilities with the required memory space and the required interpreter between the digital and analog computer will be available, a detour had to be found to get around the obstacle. A first thought was given to the various Analog Simulation programs for digital computers which are readily available and would perform almost the same functions as an analog machine but at a very slow pace [60,61,62,63,64, 65]. The publications on analog simulations are praising their methods, but it is obvious that such programs will never replace completely the actual analog computer performance. The analog is built to give a continuous flow of readings by measuring the output current, whereas the digital computer can only allow an intermittent output in a binary number system. This does not permit a continuous flow of information as is possible with the analog device and, for some applications, forms an essential part even though it may not have the high accuracy in its output as the digital computer.

For the developed method, it would have been an advantage to have a continuous output which could have been recorded on an x-y plotter and would have shown the entire coupler curve as a continuous flow of infor-

mation from the respective analog terminals. The digital computer is only able to trace this curve by calculating intermittent points, and, the finer the steps between those points, the more computer time will be needed to obtain the desired output. The analog computer can change the plotting speed by a simple change of a time variable and with a high speed would make it possible to reproduce the entire curve on an oscilloscope. Such a device would respond instantaneously to any changes in the mechanism dimensions. Analog computer programs for the coupler curve of a four-bar mechanism are readily available [46,47,48,50], and it would have been a small effort to modify the existing methods for the use with mechanisms in which none of the links can rotate through a 360 degree angle.

Because of all these advantages of an analog system, an attempt was made to produce a method which would get by with a minimum of disadvantages of the digital simulation and which would allow transfer to the use of a proper hybrid computer at a later date. For this reason, a simplified version of the simulation program was developed which does permit a speedier result. The developed method begins to move the link system in larger steps and reduces them as the mechanism does come closer to the desired position exactly opposite the desired accuracy point. Each step checks and corrects the positioning error until the final position within a prescribed accuracy is reached. Then, the distance between the obtained coupler point position and the desired precision point is recorded and eventually accounted to the error between the actual and the desired point position. Also, the angular position for each point is memorized for reducing the initial steps in obtaining the same point

positions during the next "run" around the coupler curve. In this form, the computer records and memorizes each "run" through a complete cycle of the coupler curve and is able to "learn from this experience" for its consecutive "run" around the same curve when some minor modifications are made. The magnitude and the type of these adjustments are automated and represent an important part of the optimization process.

Specification of the Desired Curve

Considerable thought was given to the method in which the desired curve should be specified and which would best adapt to the path generator synthesis. It appears that already the question of the curve specification points in a distinct direction.

Many curves which might be a desired object for the synthesis may not be expressed by an analytic expression and may be best represented by a series of points. For these cases, an analytic approximation could be found by using polynomials or other functions which are set up in higher orders depending on the type of accuracy desired. Such a function series can become a very long expression if a high accuracy is demanded. However, it may well be that in some cases the points, specifying the curve, are obtained from the desired positions of a machine part with special importance and with intermittent steps during a prescribed operation. In these cases it may not matter so much what the curve positions are in between the specified points. Then the precision points become of primary importance and may very well be used directly for the mechanism synthesis. Moreover, such a specification of a set of points will represent the most general form and will also include all analytic curves. The polynomial approximation of nonanalytic curves would introduce an

additional error and its computation would increase the required computer time. For these reasons it was felt to be best to use the point coordinates directly without any in between approximation procedure and without any long expressions for a function series. In this way, it became necessary that the developed method will not set any limit to the number of specified precision points.

The developed procedure will take any number of precision points and will check out each point for the distance between the desired and the actual obtained positions of the coupler point. It is obvious that it will take more time to check a larger number of such points compared to say only nine points. Therefore, it appears best that, if those nine points can be set up in appropriate intervals, they should be used for the beginning of the synthesis process to obtain a speedier performance of the computer and finally, the number of points can be increased stepwise to the total number of points of the entire path. This routine may require a modification to suit the individual application.

For the investigator, it was important to recognize the advantages and disadvantages of an analytical representation of the desired coupler curve at an early stage. Once it was found that the representation by a set of points would be the most general and best adaptable one, an investigation of this principle of the synthesis was made and it was found that the proposed hybrid method would also be most suitable to this form of curve representation.

Error Definition

There are various publications with definitions for the error between the desired and the actual obtained curve, and it appears that none

of the presented definitions conform to the need of an arbitrary application.

A paper by Fox [40] suggests the use of the integral

$$E = \int_a^b (y - y_g)^2 \cdot dx \quad (9)$$

representing the square of the area between the desired and the obtained curve (Figure 8). This representation may become useful for an explicit function representation of the path but becomes inconvenient for the proposed point by point representation. Therefore, the above paper proposed the use of a summation instead of the first integral.

$$E = \Delta x \cdot \sum_{i=1}^n (y_i - y_{gi})^2$$

For the curve shown in Figure 9, it may become difficult to find a good expression for the interval width Δx , especially when the curve should follow the direction of the y-axis. In such cases an arbitrary interval choice for Δx may distort the obtained result quite badly, although it might be effective for curves which are following more or less the direction of the x-axis. The squaring of the difference of the y components is needed to eliminate the negative areas between the curves. It appears that, because of the involved inaccuracy, the paper by Fox suggests that the constant value Δx be changed to a weight factor for the evaluated error in comparison to other errors like the angular position of the mechanism or factors which may be a desirable feature in the synthesis procedure.

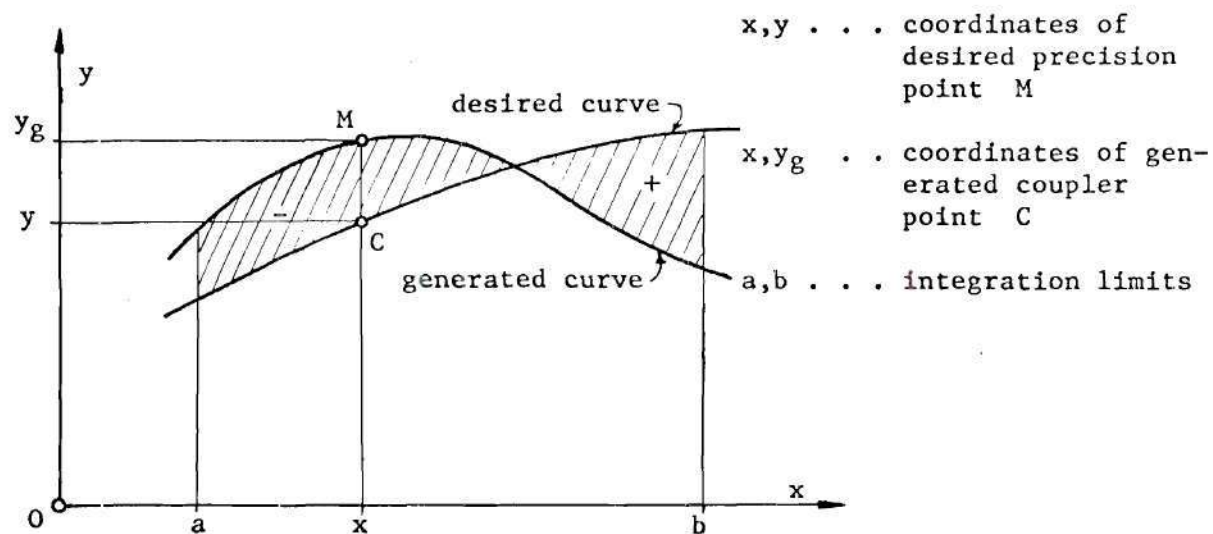


Figure 8. Integration of Error

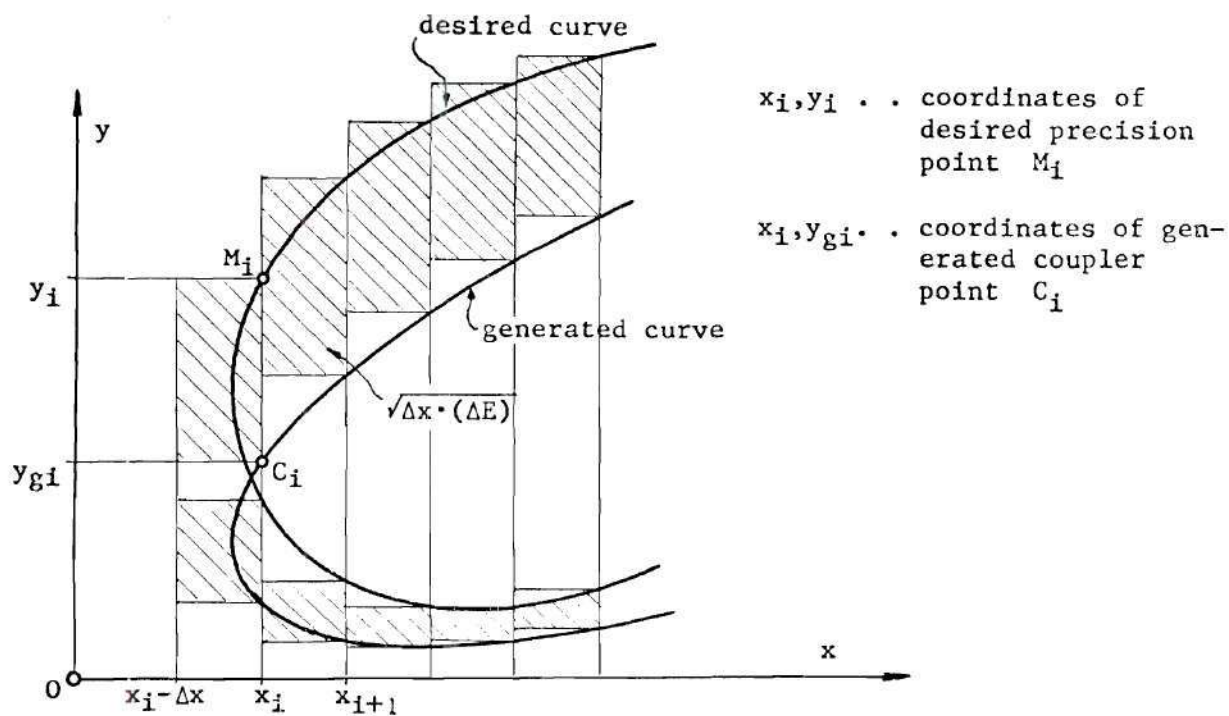


Figure 9. Summation Substitution for Error Integral

Another publication, by Lewis [66], proposes the use of an expression

$$E = \sum_{i=1}^n [(x_{gi} - x_i)^2 + (y_{gi} - y_i)^2]$$

in which both sets of coordinates, the ones of the desired and of the generated points, are depending on a parameter λ . This method is equally good for any direction of the curve but it actually does account for the hypotenuse of the error triangle as shown in Figure 10 and, due to the selected parameter λ , may yield a much bigger value than the true distance between the desired and the generated curves. The parameter λ , which brings the points in a relation to each other, may become difficult to handle and adds a restriction to the synthesis which may hinder the procedure to converge to the desired best solution. The above mentioned paper presents this added restriction as a feature of the proposed synthesis which will make it possible to preselect an angular link position for each precision point. It should be admitted that this restriction may be of some value in certain cases but does not allow one to find the most general and best solution to a given path generation problem. It is clear that the best match of a given curve can only be obtained if no restrictions are implied to the problem.

A similar method was suggested by Suh [45] in his paper which shows examples with five precision points but only indicates methods for as many as nine precision points.

An earlier paper published by Freudenstein [39] suggested the use of the height in a triangle derived from the coordinates of a given pre-

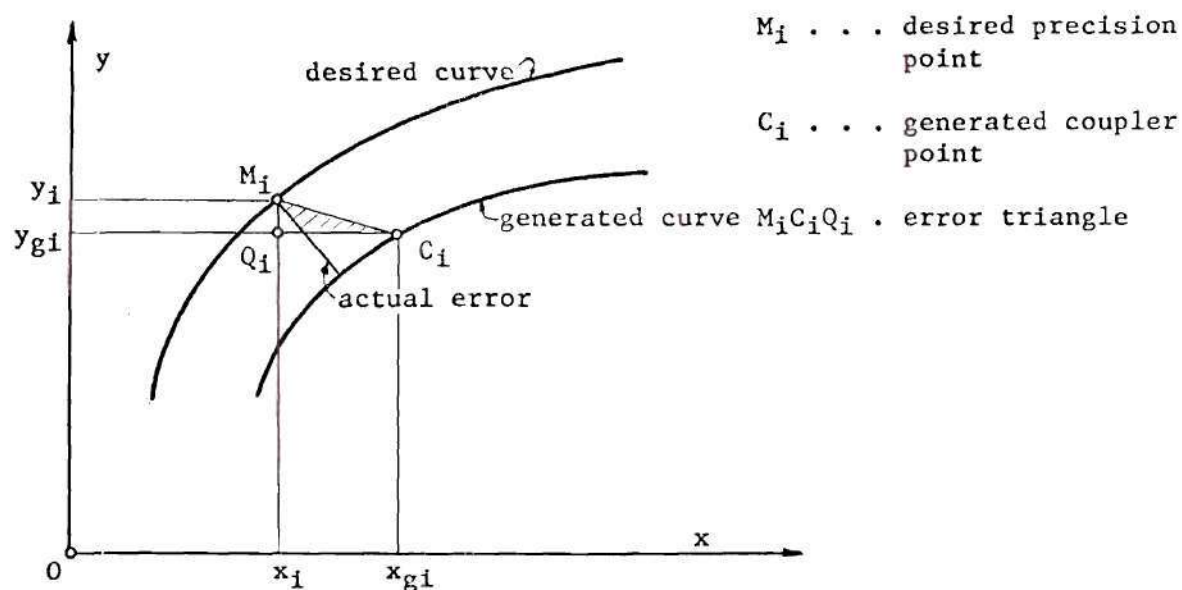


Figure 10. Error Definition Depending on a Parameter

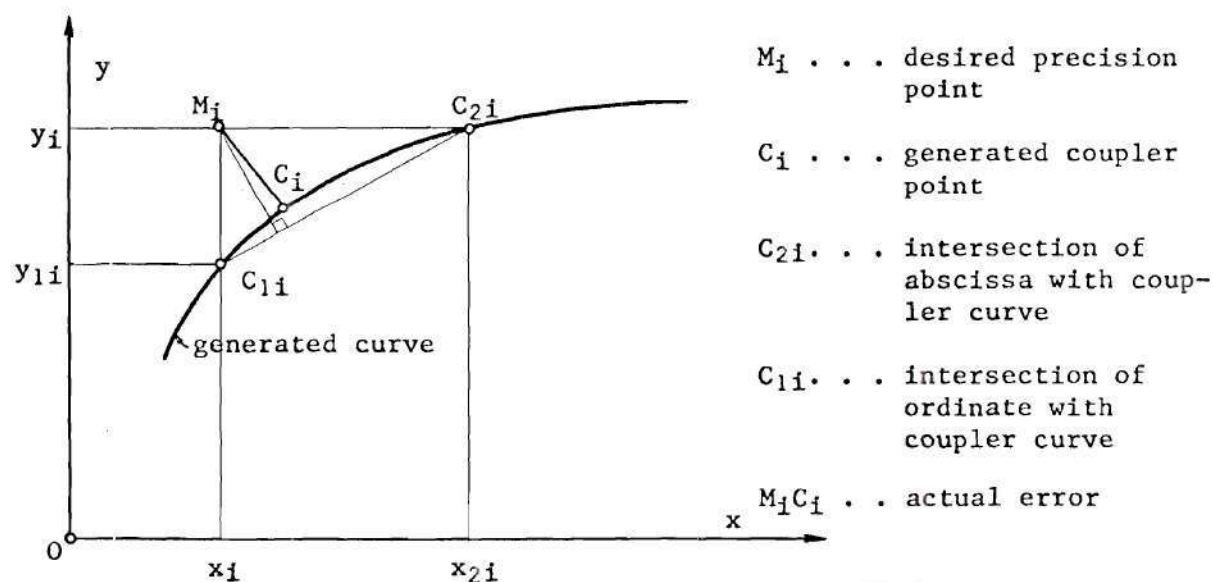


Figure 11. Error Definition with Intersection Points

cision point as shown in Figure 11. The error D can be calculated from the equation

$$\frac{1}{D^2} = \frac{1}{(x_2 - x)^2} + \frac{1}{(y_1 - y)^2}$$

in which x_2 and y are the coordinates of the intersection point between the line $y = \text{const}$ through the precision point M_i and the generated curve, and x and y_1 are the coordinates of the intersection between the line $x = \text{const}$ and the curve. Figure 11 shows clearly that with this method a poor approximation of the error will be obtained for curve areas with larger curvatures. And worse than that, it may happen that certain positions of the point M_i will not have any intersection points C_{1i} and C_{2i} with the generated curve and the results can become misleading answers when x_2 or y_1 , or both increase to a very large value in the above equation.

Consideration was given to all these methods with an attempt to overcome their shortcomings. The integral method seemed to be most attractive but could not be carried out because of the previous decision to specify the desired curve by a larger number of precision points and not by an approximating function. This implied seeking for a summation of the errors at each precision point and taking an accurate account of all perpendicular distances between the desired and the obtained curve. One also had to decide whether those distances should be taken perpendicular to the desired curve or perpendicular to the generated curve. With the given curve definition as a set of points, it was simpler to use a direction perpendicular to the generated curve. This, together

with the intended application of an analog computer, provided the guide lines which lead to the definition of the perpendicular distance.

If one was able to obtain from the analog computer the coordinates of the coupler point tracing the generated coupler curve, then it must also be possible to obtain the time derivatives of these coordinates, \dot{x} and \dot{y} , which define the direction of the tangent but also of a perpendicular line. Using the angle β of this line with the positive x-axis and positive in the CCW direction, the equation for β becomes

$$\operatorname{tg} \beta = - \frac{\dot{x}}{\dot{y}} \quad (10)$$

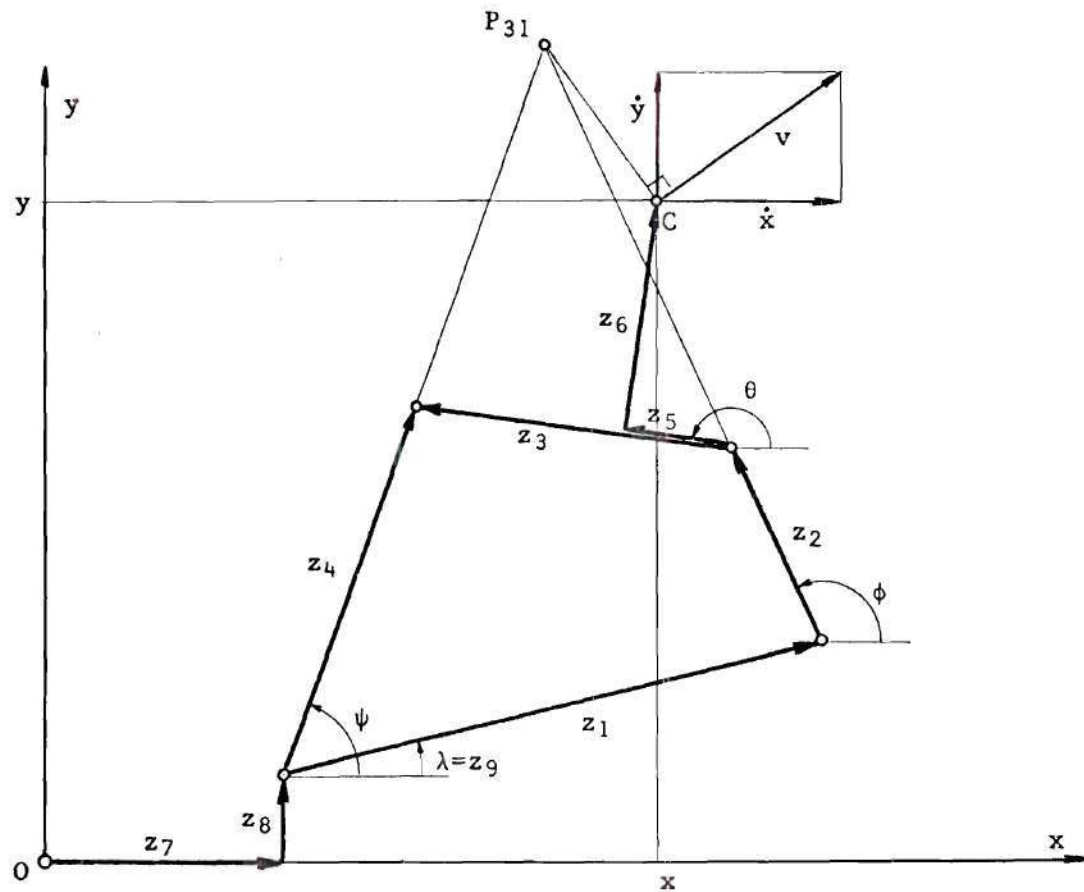
This is a simple problem for the analog computer but was not quite that simple for the digital simulation program.

If the notation as shown in Figure 12 is used for the link dimensions and the angles, it is possible to derive the following two equations

$$x = z_7 + z_1 \cdot \cos \lambda + z_2 \cdot \cos \Phi + z_5 \cdot \cos \theta + z_6 \cdot \sin \theta \quad (11)$$

$$y = z_8 + z_1 \cdot \sin \lambda + z_2 \cdot \sin \Phi + z_5 \cdot \sin \theta - z_6 \cdot \cos \theta \quad (12)$$

All link dimensions and angles are measured positive in the indicated directions. The dimensions for z_1 , z_2 , z_3 , and z_4 are limited to positive values because they will change their angular directions if these links should change to the opposite direction. For such situations, the angles λ , Φ , θ , and ψ will change their values by 180 degrees. The equations (11) and (12) can be differentiated to the time and will render the two velocity components \dot{x} and \dot{y} of the coupler point which represent at the same time the direction of the tangent to the coupler curve or the path



z_1 . . . frame

z_2 . . . crank

z_3 . . . coupler

z_4 . . . follower

z_5, z_6 . position of point
C in moving plane

z_7, z_8 . position of mechanism
in fixed plane

z_9 . . . angular position of
mechanism

NOTE:

All positive directions as
indicated.

ϕ . . . crank angle

θ . . . coupler angle

ψ . . . follower angle

λ . . . position angle

v . . . velocity direction
of coupler point

x, y . . coordinates of
coupler point

\dot{x}, \dot{y} . . velocity components

C . . . coupler point

P_{31} . . momentary pole of
coupler link

Figure 12. Notation Used in Analog Simulation

direction of the point C at any given instant.

$$\dot{x} = -z_2 \cdot \dot{\Phi} \cdot \sin \Phi - \dot{\Theta} \cdot (z_5 \cdot \sin \theta - z_6 \cdot \cos \theta) \quad (13)$$

$$\dot{y} = z_2 \cdot \dot{\Phi} \cdot \cos \Phi + \dot{\Theta} \cdot (z_5 \cdot \cos \theta + z_6 \cdot \sin \theta) \quad (14)$$

These two forms substituted into equation (10) will make it possible to evaluate the direction β of the perpendicular for any point on the coupler curve. The needed relations between the angles θ and Φ are found from the mechanism configuration in relation to the input angle Φ . It was found that the angular velocities $\dot{\Phi}$ and $\dot{\Theta}$ presented some problems which are not mentioned in reference [46].

The equations of closure for the link z_1 , z_2 , z_3 , and z_4 take the forms

$$z_1 \cdot \cos \lambda + z_2 \cdot \cos \Phi + z_3 \cdot \cos \theta = z_4 \cdot \cos \psi \quad (15)$$

$$z_1 \cdot \sin \lambda + z_2 \cdot \sin \Phi + z_3 \cdot \sin \theta = z_4 \cdot \sin \psi \quad (16)$$

and their differentiation becomes

$$-z_2 \cdot \dot{\Phi} \cdot \sin \Phi - z_3 \cdot \dot{\Theta} \cdot \sin \theta = -z_4 \cdot \dot{\psi} \cdot \sin \psi \quad (17)$$

$$z_2 \cdot \dot{\Phi} \cdot \cos \Phi + z_3 \cdot \dot{\Theta} \cdot \cos \theta = z_4 \cdot \dot{\psi} \cdot \cos \psi \quad (18)$$

Eliminating $z_4 \cdot \dot{\psi}$ from equations (17) and (18) yields

$$\begin{aligned} & z_2 \cdot \dot{\Phi} \cdot (-\sin \Phi \cdot \cos \psi + \cos \Phi \cdot \sin \psi) \\ & + z_3 \cdot \dot{\Theta} \cdot (-\sin \theta \cdot \cos \psi + \cos \theta \cdot \sin \psi) = 0 \quad \text{or} \\ & \dot{\Theta} = -\dot{\Phi} \frac{z_2}{z_3} \cdot \frac{\sin(\Phi - \psi)}{\sin(\theta - \psi)} \end{aligned} \quad (19)$$

This equation would allow one to calculate the angular velocity $\dot{\Theta}$ for a

selected input velocity $\dot{\Phi}$, used in equations (13) and (14), and would allow one to complete the evaluation of the desired angle β .

It was found that it is best to use a coordinate system which is parallel to the link z_1 with a zero angle for λ . This helps to avoid misconceptions of the required angle differences in various Boolean expressions. It, therefore, was felt to be best to transfer the coordinates each time the analog simulation is needed in the computer procedure.

Further, it was found that equation (19) had an infinite large value for $\dot{\theta}$, for θ equal to ψ . Analytically, it seemed to present no problems but in the computer no infinite large numerical quantities can be represented. Once the largest number of the computer is reached, the machine will shut off the computations. Besides, any error in the calculation of the sinus function in the denominator will amplify tremendously to a large error in the final value for $\dot{\theta}$. It was clear that the values of $\dot{\Phi}$ had to be reduced for these situations and had to approach zero for the critical position. Additionally, a different computation method for such positions of the mechanism had to be sought.

Instead of eliminating $\dot{\psi}$, one could eliminate $\dot{\Phi}$ from equations (17) and (18) and derive the form

$$\dot{\theta} = \dot{\psi} \cdot \frac{z_4}{z_3} \cdot \frac{\sin(\Phi - \psi)}{\sin(\Phi - \theta)} \quad (20)$$

which for the extreme case reduces to the form

$$\dot{\theta} = \dot{\psi} \cdot z_4 / z_3$$

because then $\sin(\Phi - \psi)$ becomes equal to $\sin(\Phi - \theta)$.

But it is also possible to find the numerical ratio \dot{x}/\dot{y} more di-

rectly and to eliminate the use of the values $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ entirely.

The sketches of Figures 13 and 14 show the two possible cases in which the $\sin(\theta-\psi)$ vanishes. It can be seen that in both of these cases the momentary pole P_{31} is at the end of link z_2 . This link will go through a period of zero angular velocity while link z_3 will rotate about the pole P_{31} , which is identical to the pin between the links z_2 and z_3 . For these two configurations, the sought tangent direction can simply be found as a line perpendicular to the connection between C and P_{31} . These components of the sought direction become

$$\begin{aligned}\dot{x} &= z_5 \cdot \sin \theta - z_6 \cdot \cos \theta \\ \dot{y} &= -z_5 \cdot \cos \theta - z_6 \cdot \sin \theta\end{aligned}$$

and the ratio for the perpendicular direction is

$$\operatorname{tg} \beta = -\dot{x}/\dot{y} = \pm \frac{z_5 \cdot \sin \theta - z_6 \cdot \cos \theta}{z_5 \cdot \cos \theta + z_6 \cdot \sin \theta} \quad (21)$$

The plus or minus sign has to be selected in reference to the rotation sense of link z_4 and the position of link z_3 . The stretched position of Figure 13 does require the plus sign and the folded position of Figure 14 the negative sign for a positive rotation of link z_4 . With a few Boolean expressions, it is possible to establish the sign in the equation for the angle β . It is interesting to notice that the same result could be obtained from dividing equation (13) by equation (14) and setting $\dot{\phi}$ equal to zero. This agrees completely with the observations made above which indicated a period of rest for link z_2 .

Attention was given to another item which is caused by the nature

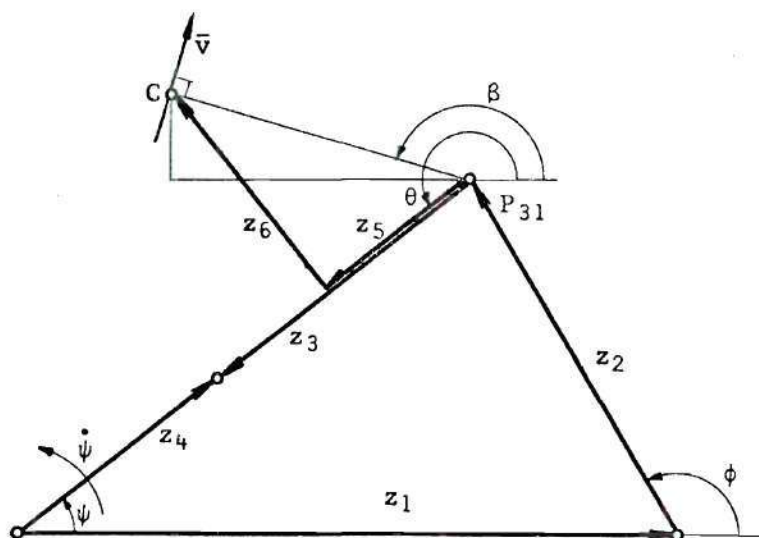


Figure 13.

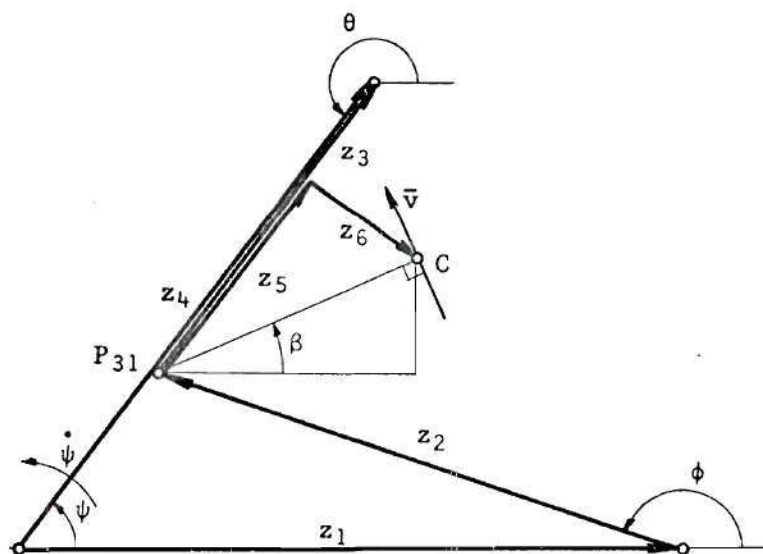


Figure 14.

Dwell Positions for Link z_2

of any angular displacements. Both arguments of the sinus functions in equations (19) and (20) are differences between two angles. Figure 15 shows $\theta - \psi$ for a $\lambda < \pi$. Figure 16 shows the same mechanism but rotated to a position in which $\lambda > \pi$ and with the relative link positions kept the same. One can observe that the value for $\theta - \psi$ became negative and the complementary angle to 360 degrees. This change has no effect on the end result of the sinus function, but it may cause difficulties in the various logical expressions used throughout the computation. This shows how important it was to bring the coordinate system into a position in which the positive x-axis became parallel to the link z_1 .

Once the ratio of \dot{x} to \dot{y} is established, it is possible to write the equation for a line normal to the coupler curve at any arbitrary position of the point C.

$$\dot{x}_i \cdot (x_i - x) + \dot{y}_i \cdot (y_i - y) = 0$$

Here x_i and y_i are the coordinates and \dot{x}_i and \dot{y}_i the velocity components of the i^{th} position of the coupler point C. The values x and y are the coordinates of a variable point on the normal line itself. It should be observed that the left side of this equation is equal to zero and it, therefore, does not matter whether the values for \dot{x}_i and \dot{y}_i are the correct values as long as the proper ratio \dot{x}_i/\dot{y}_i is maintained. If this equation is normalized and if the variables x and y are replaced by the coordinates of a precision point, it becomes

$$\dot{x}_i \cdot (x_i - x_o) + \dot{y}_i \cdot (y_i - y_o) = p \cdot \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \quad (22)$$

With p representing the perpendicular distance from the precision point

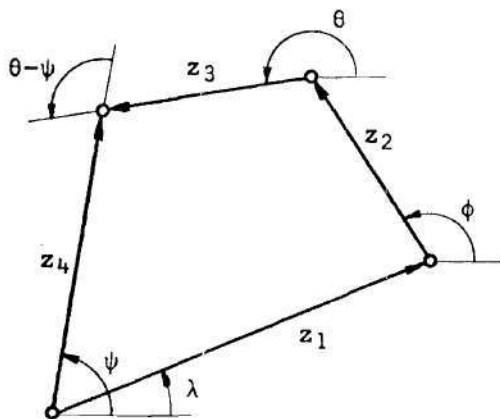


Figure 15.

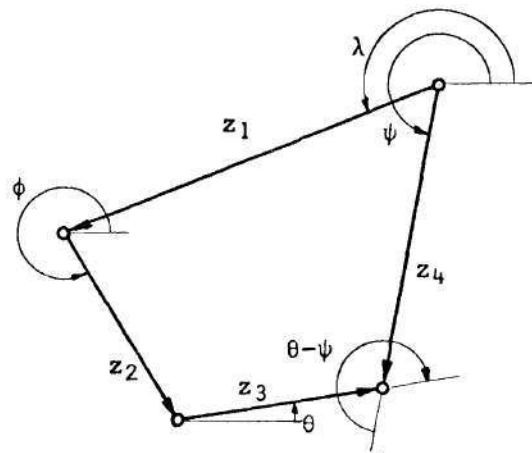


Figure 16.

Inversion of Angle Differences

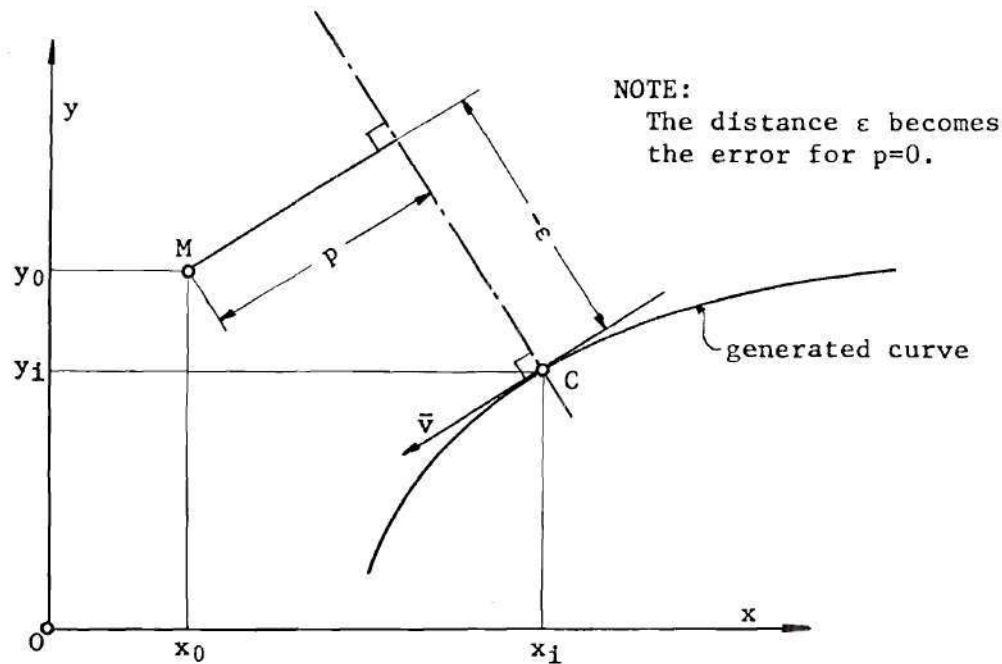


Figure 17. Line Normal to Coupler Curve

M to the line normal to the coupler curve and through the coupler point C as shown in Figure 17. The distance p will always be positive if it is on the side in which the positive velocity vector \bar{v} of the point C is pointing. The developed process moves the point C along the generated curve, preferably in a CCW direction and stops at the positions in which p becomes reasonably small (in this case 10^{-8}). For those positions, the points C and M must be on a line normal to the coupler curve, and the distance between those two points is given by

$$\epsilon_i = \pm \sqrt{(x_o - x_i)^2 + (y_o - y_i)^2} \quad (23)$$

This equation produces the positioning error ϵ and will be set positive for any ϵ with M on the left hand side of the curve looking in the direction of the motion vector \bar{v} and will be set negative for M on the other side. The total error for the entire generated path can then be expressed by

$$e = \sum_{i=1}^n |\epsilon_i| \quad (24)$$

in which n is the number of specified precision points.

Analog Simulation

The desired process was a procedure which would produce the value of the errors ϵ for all given points together with a sign indicating whether the detected error is to the left or to the right hand side of the generated curve. It, therefore, became impractical to simulate the complete operation of an analog computer and attention was given only to the problem of obtaining a listing for the errors associated with each

individual precision point. Hence, the problem setting was not to simulate the operation of an analog computer but to obtain all so called "matching errors" with their signs for a particular four-bar mechanism with a fixed orientation to the given set of precision points. The evaluation of these results for obtaining the needed changes of the physical mechanism dimensions was part of another section in the computer program.

The required input data are: the dimensions of the mechanism and a list of precision points given in the sequence which the coupler curve has to follow. All of these positions for the specified precision points should be reached by the tracer point C without the need of any disassembly of the mechanism. This required the program to take the coupler point through an entire closed section of the coupler curve with all its possible nodes and cusps.

The procedure can be called and used in the program at convenience. It will use the given coordinates of the precision points along with the last physical dimensions and orientation of the mechanism with its initial crank position, and it will render the errors for each precision point along with its sign. The procedure itself does appear rather long but consists actually of a series of sections headed by logic expressions which allow the computer to select the needed sections and to eliminate those parts which become important for the momentary mechanism configuration. In this form the computer will use only about one half of the listed statements, which will save a considerable amount of time for the process.

The positioning of the coupler point C such that the normal line

goes through the precision point M is based on a so called "regula falsi" method and was modified for this application. The value for the quantity p of equation (22) and Figure 17 is computed and indicates the distance of the normal line to the precision point under consideration. The initial position of the normal line should not be too far from the actual desired position, which is reached when p becomes zero and the normal meets the required precision point. It should always be possible to reach such a position by changing the angle Φ of the input link. The mechanism is preferably started at a point in which Φ needs to be increased to reach the desired position. Figure 18 shows a diagram of the relation $p = f(\Phi)$. The basic equation of the regula falsi method for the variables p and Φ is

$$\Phi_{i+1} = \Phi_i - (\Phi_i - \Phi_{i-1}) \cdot \frac{p_i}{(p_i - p_{i-1})}$$

or

$$\Delta\Phi_{i+1} = - \Delta\Phi_i \frac{p_i}{\Delta p_i} \quad (25)$$

and it is not difficult to see from Figure 18 that this equation will produce a negative $\Delta\Phi$ for the diagram points $p_i = p_1$ and $p_{i-1} = p_0$. This situation does occur very often and means that the value of p has to step first over a maximum before it can reach the desired value for Φ . Figure 19 shows a typical situation on the actual generated coupler curve in which the distance p is positive and does increase first before reaching the desired position C^* . It also shows that by moving C in the negative direction it will result in a vanishing p but with a much larger error. When studying the last two sketches, one can see that the sign of $\Delta\Phi_{i+1}$ in equation (25) should depend on the sign of p_i only. This change would modify the equation to the form

$$\Delta\Phi_{i+1} = |\Delta\Phi_i| \cdot \frac{p_i}{|\Delta p_i|} \quad (26)$$

and will eliminate the outlined difficulty but will not render a fool proof result for the case in which M might have the position of M'. For that case, p_0 and p_1 would become negative and the calculated $\Delta\Phi$ would also become negative and one would still obtain an undesired motion of the point C and finally end up with the wrong error reading. For such rare cases, special attention was needed which requires some extra logic expressions tailored to suit the individual application.

Another problem was the need to determine the initial step size for Φ until the first negative value for p was approached. The diagram, Figure 18, shows the first $\Delta\Phi$ between the values Φ_0 and Φ_1 , which is selected as a first guess, then the calculated value for $\Delta\Phi_2$ from equation (26) will become much too large and may finally lead to a wrong position of C*. Therefore, a check was needed for each computed step and if found too large it will be reduced to a reasonable amount. The permissible maximum step size can vary from one synthesis problem to another and needs to be adjusted to the prevailing requirements.

A further problem presented the sudden change of the slope in the curve $p = f(\Phi)$ which, for example, will cause a too large step between Φ_2 and Φ_3 , shown in Figure 18. A considerable reduction in the number of approximation steps would result if the position of Φ_3 were found closer to Φ_4 . Therefore, it appeared important to find the cause of such sudden curvature changes and to develop an improved equation (26) for $\Delta\Phi$ which should be kept simple in its construction and avoid excessive computer time. In general, every computer statement of this section had to be

weighed carefully for its importance against the needed processor time, because they are the most often used statements of the entire program. Therefore, preference was given to the use of some variable which became available in a previous section of the simulation procedure. This consideration should also explain why no attempt was made to find an exact analytic expression for the relation between the variables p and Φ . It was expected that, if such a relation can be found, it would be rather sophisticated and far more complicated to evaluate in comparison to the simple regula falsi method.

The observation of the general motion of the coupler point C revealed that basically it depends on two movements. The first one, given by the angular velocity $\dot{\Phi}$ indicating the rotation of the link z_2 , and the second one given by a relative rotation indicating the motion of the link z_3 in respect to link z_2 . If the angle Φ is directly related to the time and $\dot{\Phi}$ kept constant, then the variation of the relative velocity becomes a function of $\dot{\theta}$ only. $\dot{\theta}$ was previously calculated for the directions \dot{x} and \dot{y} . It was further found that an increased or reduced $\dot{\theta}$ meant a larger change of p for a certain value of $\Delta\Phi$ calculated from equation (26). In other words, it appeared that the sudden change of the slope of the curve in Figure 18 was caused by the changing value $\dot{\theta}$. With this recognition, it was not too complicated to introduce a damping factor which would reduce the calculated values for $\Delta\Phi_{i+1}$. The expression

$$\left| \left| \frac{\dot{\theta}_i}{\dot{\theta}_{i-1}} \right| - 1 \right|$$

would always remain positive but would become zero if there was no change in $\dot{\theta}_i$ against $\dot{\theta}_{i-1}$. This simple form added to the denominator will reduce

the calculated value of $\Delta\phi_{i+1}$ and the new equation (26) can be constructed as

$$\Delta\phi_{i+1} = |\Delta\phi_i| \cdot \frac{p_i}{|p_{i-1} - p_i| + \left| \left| \theta_i / \theta_{i-1} \right| - 1 \right|} \quad (27)$$

and was found sufficiently effective in its general response.

This simulation procedure is called for each precision point and will basically follow the following steps.

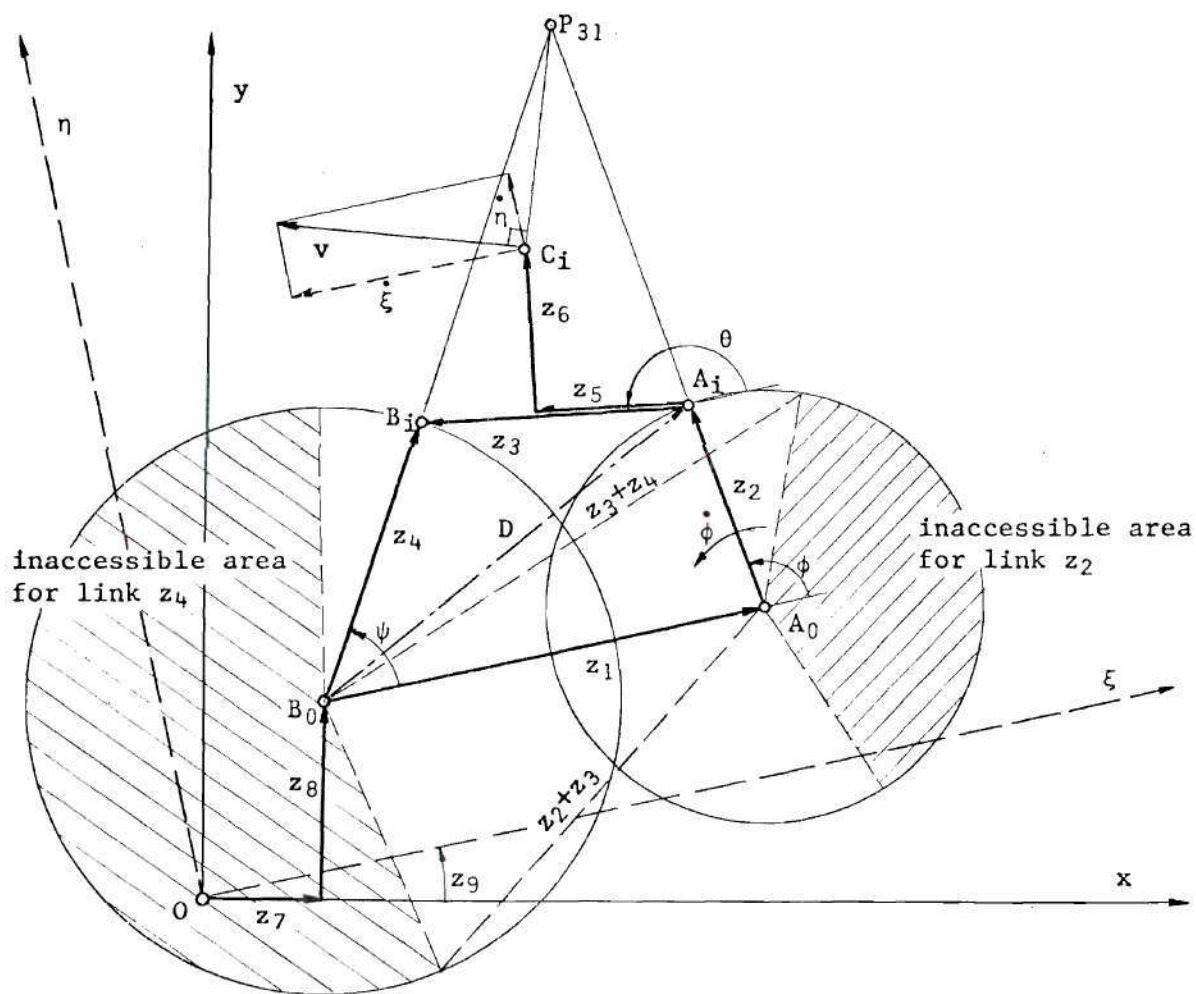
1. It will change the coordinates of the precision point M to a new coordinate system which is in line with the positive direction of the link z_1 and all angles are measured in relation to these new coordinates, ξ and η .
2. It will check the initial angle ϕ for its position which should be compatible with the mechanism motion.
3. The angles ψ_i and θ_i are calculated for their exact values from the closure equations which read in complex form for the i^{th} position (Figure 20).

$$\begin{aligned} z_4 \cdot e^{i\psi_i} &= z_1 + z_2 \cdot e^{i\phi_i} + z_3 \cdot e^{i\theta_i} \\ D_i &= z_1 + z_2 \cdot e^{i\phi_i} \end{aligned} \quad (28)$$

The angles are evaluated in a manner similar to the one mentioned in chapter III and a choice between the symmetry angles is made on the ground of the magnitude of the previous angle ψ , fetched from the computer memory.

4. The values for $\dot{\xi}$, $\dot{\eta}$, ξ , and η are calculated including the required $\dot{\theta}$ based on a constant $\dot{\phi}$.

5. The distance p is calculated and checked for its size. If



z_1 . . . frame
 z_2 . . . crank
 z_3 . . . coupler
 z_4 . . . follower

z_5, z_6 . vectors positioning coupler point in moving plane
 z_7, z_8 . vectors positioning mechanism in fixed plane
 z_9 . . . angular position of mechanism

A_0, B_0 . fixed pins
 A_1, B_1 . moving pins
 C_1 . . . coupler point
 P_{31} . . momentary pole
 x, y . . fixed coordinates
 ξ, η . . rotated coordinates

Figure 20. Rotation of Coordinates about the Angle z_9

found small enough the computer will exit the procedure by computing the error ϵ and its sign. This is possible by the use of the values obtained last for $\dot{\xi}$, $\dot{\eta}$, ξ , and η from the generated curve together with the modified coordinates for the precision points ξ_0 and η_0 .

6. If p was not zero and was calculated for the first time, then an estimated $\Delta\phi$ which entered the procedure is used, otherwise a new $\Delta\phi$ is calculated by using the modified "regula falsi" method.

7. The processed $\Delta\phi$ is checked for its compatibility, that is whether or not it would change the angle ϕ to a value which forces link z_2 into an inaccessible position (see Figure 20).

8. Step 3 is repeated under the modified conditions with a new ϕ which was obtained by adding the last $\Delta\phi$.

9. The steps 4 down to 8 are repeated until p becomes sufficiently small and an exit can be made in step 5.

These nine steps of the procedure were quite effective for standard linkage systems and were found satisfactory until some special cases of double rocker mechanisms and others were tried. First it was found that the loop formed by the steps 4 to 8 may not be exited by some unforeseen reason and may cause the remaining computer time to go to waste. Under such conditions, a forced exit was found to be of great help and saved the remaining time for some further computations. In many cases, a single positioning mistake will correct itself during the next call for the procedure. For this reason, a time stop was added which will cause the exit if the loop runs too long.

In addition, it was found that for certain positions of link z_2 it became almost impossible to obtain a proper convergency to a zero p

condition. These cases were found to be in areas where the momentary pole of the coupler link z_3 approached the location of the pin between z_2 and z_3 as shown in Figures 13 and 14. That is the case for which the denominator of equation (19) becomes infinitely small. For these approaching situations, it was found best to give the control of the mechanism motion to the angle ψ and to make the link z_4 the driver. That meant to change the role of the driver with the driven link until the critical position is past.

CHAPTER V

MATRIX INVERSION

General Comments

A paper published by Greville [67] did apply a matrix notation to the least square method for the solution of a set of linear equations. There were either more or less unknowns than equations. For this investigation, only the case of more equations than unknowns was of special interest. A further improvement represented the method developed by Diederich [68] which does allow a grouping of the equations. The first group which has to consist of fewer equations than unknowns, or independent variables, will be solved exactly and the second group formed by the remaining equations is approximated by the least square. In this form one can put more emphasis on some equations in the program and let the others be approximated as good as possible. Because each equation represents the result of one precision point and because there are as many equations as precision points, it became possible to put more emphasis on the synthesis of some important precision points.

In this analysis, there are nine independent variables responsible for the dimension of the four-bar mechanism and they are to be modified to an optimum solution. The number of precision points is assumed to be larger than nine. However, if only nine points are needed, the outlined procedure will reduce itself to an ordinary matrix inversion after checking the rank of the matrix. It is hard to believe that finding a dependency in the columns or rows of the matrix ever may happen.

But, there were so many unexpected events during this investigation that it was felt to be an asset to have this extra check in the matrix form of the least square method which then would work for any dependency case.

It is not difficult to understand that the number of equations of the first group, which is to be solved exactly, cannot exceed nine. With nine equations in the first group, it is obvious that the remaining equations cannot be approximated, because the first group will become a straightforward matrix inversion and will not leave any choice of independent variables for an approximation of the second group. In this manner, one can also understand that the general approximation of the second group will become a better one for a smaller number of equations in the first group and is best if this number approaches zero. Obviously, this last case will bring the procedure back to the ordinary least square inversion method.

Matrix Inversion

The inversion of a square matrix can be accomplished by the Gauss method, independent of the number of equations and variables, provided the determinant of the matrix is non-zero.

The usual algorithm requires two loops in its computer program. Here a shorter method needing only one loop will be described.

In the basic two-loop procedure, the first step is to transform the square coefficient matrix into a triangular one, and the second step is needed to evaluate the unknown variables x beginning with the last one, x_n , and ending with the first one, x_1 . This is well known. When the triangulation is completed, the last equation has only a single variable and can be solved easily for its value. Here begins the second step

with the calculation of x_n and, working backwards, the next unknown, x_{n-1} , can be found from the second last equation, etc. This latter process cannot be combined with the triangulation of the matrix and normally has to be programed in a separate loop. The new procedure used for this matrix inversion was developed in order to avoid disappointments with library procedures, which sometimes do not perform as hoped, because it is difficult to build into those standard procedures all possible applications. The experience shows that one can increase the effectiveness of such procedures if they are designed for a particular job only. Moreover, the designer becomes very versatile in its application and can introduce any needed modifications without difficulties.

The applied one-loop algorithm actually serves more than one purpose. It will find the solution to a set of inhomogeneous equations, it will invert a matrix and at the same time will establish the rank of the matrix. Finally, it will tell which of the equations cause a dependency if the rank came out smaller than the number of equations.

A. One-loop Method for the Solution to a Set of Inhomogeneous Equations

Consider the set of linear equations in matrix form

$$A \bar{x} = \bar{a} \quad (29)$$

where A is a $n \times n$ matrix, and \bar{x} and \bar{a} are column vectors. If it is conceivable to transform A into the unit matrix I , then the right hand side would equate to \bar{x} and will represent the sought solutions.

$$I \bar{x} = \bar{c} \quad \text{or} \quad \bar{x} = \bar{c} \quad (30)$$

The used transformation is similar to the Gauss method but, instead

of leaving the matrix in a triangular form, the transformation is carried on until a unit matrix is obtained. In order to simplify the representation, some new line vectors \bar{s} are introduced which comprise all coefficients of one equation with an additional factor at the right hand side representing the respective value of the vector \bar{a} . The line vector \bar{s} has super and sub scripts to indicate the stage of transformation and the represented line.

$$\begin{array}{ccccccc}
 & 1 & 2 & 3 & & r & & n & \bar{a} \\
 & \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\
 {}^0\bar{s}_1 & = & a_{11} & a_{12} & a_{13} & \cdot \cdot & a_{1r} & \cdot \cdot & a_{1n} & a_1 \\
 {}^0\bar{s}_2 & = & a_{21} & a_{22} & a_{23} & \cdot \cdot & a_{2r} & \cdot \cdot & a_{2n} & a_2 \\
 {}^0\bar{s}_3 & = & a_{31} & a_{32} & a_{33} & \cdot \cdot & a_{3r} & \cdot \cdot & a_{3n} & a_3 \\
 \cdot & & \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\
 \cdot & & \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\
 {}^0\bar{s}_n & = & a_{n1} & a_{n2} & a_{n3} & \cdot \cdot & a_{nr} & \cdot \cdot & a_{nn} & a_n
 \end{array} \tag{31}$$

The first step is a simple transformation of the first column elements to obtain a "one" in position 11 and zeros in all others which then represent the first column vector of the desired unit matrix I. Using the line vectors and its representation in long forms, this transformation will produce a new form in which ${}^0s_{21}$ stands for the first coefficient of the second line vector, that is the position 21 in the coefficient matrix A before the transformation was performed.

difference between each row itself and the product of the respective terms of the pivot row and the first non-zero factor of the row under transformation.

Now it is possible to show the general transformation law which will allow one to transform the matrix A into a unit matrix I. If the row numbers are given by the index i and the transformation number by the index k, it can be presented by the two equations

$$\bar{s}_i^k = \bar{s}_i^{k-1} / s_{ii}^{k-1} \quad \text{for } i = k \quad (32)$$

$$\bar{s}_i^k = \bar{s}_i^{k-1} - s_{ik}^{k-1} \cdot \bar{s}_k^{k-1} \quad \text{for } i \neq k \quad (33)$$

Here, k will take all values from 1 to n and i will also take all values from 1 to n for each value of k. Equation (32) must be evaluated first because its result is needed for the substitution of the last term in equation (33). After the completion of the n^{th} transformation, all rows are transferred into unit vectors with the desired unknowns shown as last terms of the row. One should recall that each row of the vector \bar{s} had $n+1$ terms because of the addition of the factors from the vector \bar{a} . This was done to make it possible to include these terms in the recursive process of the transformation. The term x_1 will be shown as the last term of the first row, x_2 as the last term of the second row, and x_n of the n^{th} row.

It is easy to program the two equations (32) and (33) with the outlined double loop, but one question remains. What should be done if one of the divisors in equation (32) becomes zero? And, there is also another matter which needs consideration in equation (33). Here a differ-

ence is taken between values which may become extremely unequal for the case that the divisor of equation (32) was very small compared to its other value. This combined situation creates a difference between a large and a small value and will produce a rather inaccurate result for the next transformation. It, therefore, is essential to rearrange the equations in a form in which the larger coefficients become located in the diagonal of the matrix. It is easily recognized that such an interchange in the vertical direction for obtaining the larger values has the advantage of not having any effect on the sequence of the terms in the resulting vector \bar{x} . Such a change has no effect on the positions of the unit vectors already established because they are excluded from the search for larger values. That is, the search will have to be limited to rows which have an index number larger than k , the number of transformation.

However, this advantage of leaving the sequence of \bar{x} undisturbed had to be discarded, because it turns out that, for those cases in which a dependency exists in the matrix A , all factors of one column including the one in the diagonal vanish at the same time; whereas some remaining factors in higher order columns (with column numbers larger than k) will remain. This situation caused the computer to stop too soon in its procedure and left the remaining columns unconsidered for its inversion, although those columns may still be a part of the matrix which had no dependency. For this reason, it became necessary to search for the largest values along the diagonal of the matrix, and a rearrangement of the vector \bar{x} has to be executed after the last transformation. Once the largest factor in the diagonal is found, its position is recorded for the

later regrouping of the terms in \bar{x} , and the rows and columns are interchanged as needed. In this form it is possible to keep dependent rows or columns to the last and to let the computer stop the procedure only after all independent terms are evaluated.

B. Matrix Inversion

Consider equation (29) again, but now for two sets of variables \bar{x} and \bar{y} and for two vectors \bar{a} and \bar{b} ; one will obtain the matrix relation

$$A [\bar{x}, \bar{y}] = [\bar{a}, \bar{b}] \quad (34)$$

The first set with the unknowns in vector \bar{x} will equate to the factors in \bar{a} and the second set of unknowns in \bar{y} will equate to \bar{b} . It can be seen that the algorithm of section A can also be used for equation (34) and will simultaneously solve the two sets of n equations. All that is needed is one additional factor in each line \bar{s} such that it will read

$${}^0\bar{s}_i = a_{i1} \ a_{i2} \ a_{i3} \ \cdot \cdot \cdot \ a_{in} \ a_i \ b_i$$

in which b_i represents the respective factor from the vector \bar{b} . After the n^{th} transformation, the equation (34) will assume the form

$$I [\bar{x}, \bar{y}] = [\bar{c}, \bar{d}]$$

and will render the desired solutions in the matrix $[\bar{c}, \bar{d}]$, each in its respective position.

It can be seen that this method, if good for two vectors, must also work for n vectors and has the general form

$$A X = B \quad (35)$$

in which also the forms X and B became $n \times n$ matrices. The transformation will again change A into a unit matrix and produce the form

$$I X = C$$

It is now interesting to find out what does happen when equation (35) has a unit matrix I instead of a general matrix B . One must remember that the transformation actually finds the product of the inverted matrix A with the matrix B . For equation (35) this would mean

$$X = A^{-1} B = C$$

If at the beginning one had, instead of B , a unit matrix I such that

$$A X = I$$

then the transformation will supply the inverted matrix of A

$$X = A^{-1} I \quad \text{or} \quad X = A^{-1}$$

However, a full transformation can only be performed if A has a non-vanishing determinant or, what is the same, if there were no dependency in the coefficient vectors of the rows or columns. It therefore becomes important to investigate those cases in which a dependency does exist.

For the case of a dependency in the columns, it is best to modify the equations (32) and (33) to a new form which uses the column vector \bar{c} instead of the line vector \bar{s} . The form ${}^k\bar{c}_i$ would represent the i^{th} column after the k^{th} transformation and the transformation law can be shown in its general form as

$${}^k\bar{c}_i = {}^{k-1}\bar{c}_i - {}^{k-1}\bar{c}_k \cdot {}^k c_{ki} \quad \text{for } i > k \quad (36)$$

This equation is only of limited value because of a different method for the transformation of the k^{th} term in each vector. This one is simply divided by the k^{th} factor in the k^{th} column.

$${}^k c_{ki} = {}^{k-1} c_{ki} / {}^{k-1} c_{kk} \quad \text{for } i > k$$

For this reason, one cannot consider equation (36) as a full vector equation. However, it is valid when excluding the terms with an index k and will help to investigate the column dependency of the matrix A .

Such a dependency is established if the relation

$$\alpha \bar{x}_1 + \beta \bar{x}_2 + \gamma \bar{x}_3 + \dots + \sigma \bar{x}_n = 0$$

exists for not all coefficients $\alpha, \beta, \gamma, \dots, \sigma$ equal to zero. The generality of this investigation will not suffer if a simpler vector dependency of the type

$${}^0 \bar{c}_{r+1} = \lambda \cdot {}^0 \bar{c}_r + \gamma \cdot {}^0 \bar{c}_{r-1} \quad (37)$$

with only two non-vanishing coefficients is chosen. The pre-index 0 refers to the original coefficients of the matrix A before any transformation was executed.

The first question which arises is whether or not the transformation has any effect on the equation (37). Excluding the first factors of each vector in equation (36) for the first transformation, it will read

$${}^1 \bar{c}_i = {}^0 \bar{c}_i - {}^0 \bar{c}_1 \cdot {}^1 c_{1i}$$

and the omitted first factors are

$${}^1 c_{1i} = {}^0 c_{1i} / {}^0 c_{11}$$

Substituting the values for ${}^1c_{1i}$ the columns i and $i-1$ become

$${}^1\bar{c}_i = {}^0\bar{c}_i - {}^0\bar{c}_i \cdot \frac{{}^0c_{1i}}{{}^0c_{11}} \quad \text{and} \quad {}^1\bar{c}_{i-1} = {}^0\bar{c}_{i-1} - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i-1}}{{}^0c_{11}}$$

Finally, adding these two equations and substituting equation (37) into it, it can be shown that

$${}^1\bar{c}_{i+1} = \lambda \cdot {}^1\bar{c}_i + \gamma \cdot {}^1\bar{c}_{i-1} = \lambda \left({}^0\bar{c}_i - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i}}{{}^0c_{11}} \right) + \gamma \left({}^0\bar{c}_{i-1} - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i-1}}{{}^0c_{11}} \right)$$

But there is also

$${}^0\bar{c}_{i+1} = \lambda \cdot {}^0\bar{c}_i + \gamma \cdot {}^0\bar{c}_{i-1} \quad \text{and} \quad {}^0c_{1i+1} = \lambda \cdot {}^0c_{1i} + \gamma \cdot {}^0c_{1i-1}$$

which allow reducing the above form to

$$\lambda \left({}^0\bar{c}_i - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i}}{{}^0c_{11}} \right) + \gamma \left({}^0\bar{c}_{i-1} - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i-1}}{{}^0c_{11}} \right) = {}^0\bar{c}_{i+1} - {}^0\bar{c}_1 \cdot \frac{{}^0c_{1i+1}}{{}^0c_{11}}$$

The same finding will be discovered if stepping from the k^{th} to the $k+1^{\text{th}}$ transformation and proves that the proposed transformation has no effect on the relations amongst the column vectors.

Considering now the $r-1^{\text{th}}$ together with the r^{th} transformation, equation (36) will have the forms

$${}^{r-1}\bar{c}_i = {}^{r-2}\bar{c}_i - {}^{r-2}\bar{c}_{r-1} \cdot \frac{{}^{r-2}c_{r-1i}}{{}^{r-2}c_{r-1r-1}} \quad \text{for } i > r-1 \quad (38)$$

$${}^r\bar{c}_i = {}^{r-1}\bar{c}_i - {}^{r-1}\bar{c}_r \cdot \frac{{}^{r-1}c_{ri}}{{}^{r-1}c_{rr}} \quad \text{for } i > r \quad (39)$$

Studying the effects of these transformations on the r^{th} and $r+1^{\text{th}}$ column it is found that, in the first case, i becomes r in equation (38) and it will read

$${}^{r-1}\bar{c}_r = {}^{r-2}\bar{c}_r - {}^{r-2}\bar{c}_{r-1} \cdot \frac{{}^{r-2}c_{r-1} r}{{}^{r-2}c_{r-1} r-1} \quad (40)$$

and the $r+1^{\text{th}}$ column vector is formed by

$${}^{r-1}\bar{c}_{r+1} = {}^{r-2}\bar{c}_{r+1} - {}^{r-2}\bar{c}_{r-1} \cdot \frac{{}^{r-2}c_{r-1} r+1}{{}^{r-2}c_{r-1} r-1} \quad (41)$$

for all terms with an index different from $r-1$. Introducing equation (37), which holds for the vectors but also for the scalar values of each vector component, equation (41) will take the form

$$\begin{aligned} {}^{r-1}\bar{c}_{r+1} &= \lambda \cdot {}^{r-2}\bar{c}_r + \gamma \cdot {}^{r-2}\bar{c}_{r-1} - {}^{r-2}\bar{c}_{r-1} \cdot \left(\gamma + \lambda \cdot \frac{{}^{r-2}c_{r-1} r}{{}^{r-2}c_{r-1} r-1} \right) \quad (42) \\ &= \lambda \cdot \left({}^{r-2}\bar{c}_r - {}^{r-2}\bar{c}_{r-1} \cdot \frac{{}^{r-2}c_{r-1} r}{{}^{r-2}c_{r-1} r-1} \right) \end{aligned}$$

Now, also applying the transformation of equation (39) to the $r+1^{\text{th}}$ column, it will yield

$$\bar{c}_{r+1} = {}^{r-1}\bar{c}_{r+1} - {}^{r-1}\bar{c}_r \cdot \frac{{}^{r-1}c_{rr+1}}{{}^{r-1}c_{rr}} \quad (43)$$

Introducing the result from equations (40) and (42) into equation (43) we have

$$\bar{c}_{r+1}^r = \lambda \cdot \left(\bar{c}_r^{r-2} - \bar{c}_{r-1}^{r-2} \cdot \frac{c_{r-1, r}^{r-2}}{c_{r-1, r-1}^{r-2}} \right) - \left(\lambda + \gamma \cdot \frac{c_{rr-1}^{r-1}}{c_{rr}^{r-1}} \right) \quad (44)$$

$$\cdot \left[\bar{c}_r^{r-2} - \bar{c}_{r-1}^{r-2} \cdot \frac{c_{r-1, r}^{r-2}}{c_{r-1, r-1}^{r-2}} \right] = 0$$

because $c_{r, r-1}^{r-1}$ must be zero according to equation (36), which indicates that k should not be larger or equal to i , but here i is equal to $r-1$ and k shows the $r-1^{\text{th}}$ transformation which will make $c_{r-1, r-1} = 1$ and $c_{r, r-1} = 0$. This indicates that, if a dependency according to equation (37) exists, all elements of the vector \bar{c}_{r+1} with an index higher than r will become zero after the r^{th} transformation. Because of the different transformation law of the r^{th} vector element in equation (43), the result of equation (44) becomes invalid for vector elements with an index smaller than $r+1$. And, because of the same difference in the diagonal elements of all previous transformations, this result also cannot be applied to any elements with indexes smaller than r . This, however, is not important for the demonstration that all elements with higher indexes vanish for a column dependency in the matrix A . The result shows that the procedure will find a zero in the $r+1$ column and, due to the search for a maximum value in the diagonal elements, will push this column to the last position. Eventually, when there are no other non-zero elements left to carry on its transformation, the procedure will stop and all dependent vectors will be collected at the right hand side of the matrix.

For the second case of a dependency in the rows, it is somewhat easier to show which of the line vector elements will vanish. Using,

this time, the line vectors $k_{\bar{s}_1}$, a general dependency can be expressed, as before, by

$$k_{\bar{s}_{r+1}} = \lambda \cdot k_{\bar{s}_r} + \gamma \cdot k_{\bar{s}_{r-1}} \quad (45)$$

Using the transformation of equation (33) for the $r-1^{\text{th}}$ transformation, the equations for the r^{th} and $r+1^{\text{th}}$ line will change to

$$r-1 \bar{s}_r = r-2 \bar{s}_r - r-2 s_{rr-1} \cdot \frac{r-2 \bar{s}_{r-1}}{r-2 s_{r-1r-1}} \quad (46)$$

$$r-1 \bar{s}_{r+1} = r-2 \bar{s}_{r+1} - r-2 s_{r+1r-1} \cdot \frac{r-2 \bar{s}_{r-1}}{r-2 s_{r-1r-1}} \quad (47)$$

with equation (32) substituted for the last term of equation (33). Substituting equation (45) into (47) will render

$$\begin{aligned} r-1 \bar{s}_{r+1} &= \lambda \cdot r-2 \bar{s}_r + \gamma \cdot r-2 \bar{s}_{r-1} \\ &\quad - (\lambda \cdot r-2 s_{rr-1} + \gamma \cdot r-2 s_{r-1r-1}) \cdot \frac{r-2 \bar{s}_{r-1}}{r-2 s_{r-1r-1}} \\ &= \lambda \cdot \left(r-2 \bar{s}_r - r-2 s_{rr-1} \cdot \frac{r-2 \bar{s}_{r-1}}{r-2 s_{r-1r-1}} \right) \end{aligned} \quad (48)$$

Considering now the r^{th} transformation executed according to equation (33) on the $r+1^{\text{th}}$ line vector will give

$$r \bar{s}_{r+1} = r-1 \bar{s}_{r+1} - r-1 s_{r+1r} \cdot \frac{r-1 \bar{s}_r}{r-1 s_{rr}} \quad (49)$$

and substituting equations (48) and (46) and the r^{th} component from equation (48), which is

$$r^{-1} s_{r+1 r} = \lambda \cdot \left(r^{-2} s_{rr} - r^{-2} s_{rr-1} \cdot \frac{r^{-2} s_{r-1 r}}{r^{-2} s_{r-1 r-1}} \right)$$

into equation (49), it will take the form

$$\begin{aligned} r^{-1} \bar{s}_{r+1} &= \lambda \cdot \left[r^{-2} \bar{s}_r - r^{-2} s_{rr-1} \cdot \frac{r^{-2} \bar{s}_{r-1}}{r^{-2} s_{r-1 r-1}} \right] \\ &- \lambda \cdot \left(r^{-2} s_{rr} - r^{-2} s_{rr-1} \cdot \frac{r^{-2} s_{r-1 r}}{r^{-2} s_{r-1 r-1}} \right) \cdot \left[r^{-2} \bar{s}_r - r^{-2} s_{rr-1} \cdot \frac{r^{-2} \bar{s}_{r-1}}{r^{-2} s_{r-1 r-1}} \right] / \left(r^{-2} s_{rr} - r^{-2} s_{rr-1} \cdot \frac{r^{-2} s_{r-1 r}}{r^{-2} s_{r-1 r-1}} \right) = 0 \end{aligned} \quad (50)$$

This shows again that, for a dependency of the type as in equation (45), the r^{th} transformation will make all elements in the $r+1^{\text{th}}$ row with an index larger than r vanish. Due to the search for non-vanishing elements in the diagonal of the matrix, all dependent line vectors will be pushed to the bottom end and will again be recognized by the procedure as dependent vectors in the matrix.

In both types of dependencies the two elements and the column elements, respectively, with an index larger than r became zero. The search in the diagonal for non-zero elements will in this form not only sort out all dependent vectors but will also give a better accuracy by dividing consistently by the largest diagonal element.

It is necessary, however, to remember that the computer is not able to execute any calculations without a small error at the last manipulated digit. For this reason, a difference of two real numbers will never become exactly zero but will result in a very small value. To correct this deficiency and to help the recognition of dependencies, a statement was added which will recognize such small values and set them precisely to zero.

C. Recognition of Independent Section and Rank of the Matrix

It is known from the basic matrix calculus that the rank of a matrix is the number of independent row or column vectors. For the case that there are fewer independent columns than rows or vice versa, the smaller number will establish the rank of such a matrix.

In the developed inversion procedure, an additional row with an index $n+1$ was added to the elements of the matrix A which will not be included in the transformations but will only keep a record of the positions of the column numbers from 1 to n . Each time a column position is altered, these numbers will change their positions accordingly. Now it is not difficult to understand that, when the procedure comes to a halt before all n transformations are executed, the index of the last completed transformation is equal to the rank and the $n+1^{\text{th}}$ row will contain the numbers of those rows or columns which caused the halt of the transformation. They will be listed in the positions beyond those equal to the rank number. All independent vectors will show with their numbers in positions up to and including the rank number. They can be regrouped to their original sequence together with the independent vectors in the matrix. Because of the interchanging of elements in the diagonal direc-

tion, the recorded changes will affect either the rows or the columns, and the program will not be able to distinguish whether the dependency occurred in the rows or in the columns, but in this application this item has no effect on the end result.

Matrix Inversion by the Least Square Method

It is known that the least square method will not render the best possible solution if the aim is to minimize the sum of the absolute errors but that this method becomes a mathematical tool to rid the calculations of any inhomogeneity.

$$e = \sum_{i=1}^n |\epsilon_i| > \sum_{i=1}^n \epsilon_i^2 \quad \text{for} \quad \epsilon_i < 1 \quad (51)$$

$$e = \sum_{i=1}^n |\epsilon_i| < \sum_{i=1}^n \epsilon_i^2 \quad \text{for} \quad \epsilon_i > 1 \quad (52)$$

If ϵ is a function of one variable only and if it could be considered as a linear function at least for the area under investigation, the sum of the absolute values can be calculated on the computer rather rapidly and the minimum could be found by checking all critical points. In such a case the function for e is represented by a string polygon of which one of the knots forms the desired absolute minimum.

However, the analysis of this paper uses an ϵ which is nonlinear and a function of nine variables. For this situation, no method for the "absolute" minimum is known. But even if there were such a solution, it remains questionable whether or not this would help to improve the convergency of the desired optimization which is the minimization of the

absolute values for ϵ . As it shall be explained later in the chapter on relaxation, it is known that a certain amount of overrelaxation does help the speed of convergency and, therefore, the least square method may possibly provide such a condition and become the preferred method against the least absolute values. If e represents the sum of all ϵ_i taken as absolute values, then the equations (51) and (52) do show quite clearly that all ϵ_i^2 of ϵ 's smaller than one will be reduced and all ϵ_i^2 of ϵ 's larger than one will be increased in their weight which contribute to the total value of e . This distortion turns out to be an advantage for the desired convergency similar to the desirable overrelaxation. The squaring itself, however, represents only a tool to overcome the difficulty of the absolute values in the computation, because for finding a minimum for e it is necessary to find the differentials of all ϵ 's under the sum and, as it can be seen, these differentials would become discontinuous at points where one of the ϵ_i changes its sign.

Then the question arises, why should not the standard least square method be used which can be programmed without difficulties for nine unknowns and n linear equations? Yes, this is possible as long as there is no column singularity in the coefficient matrix of the linear equations. A row singularity or dependency would not disturb the process but a column dependency can cause a division by zero and would stop the process. It became necessary to eliminate such situations by establishing the rank of the matrix first and by using a certain amount of matrix calculus throughout the computations. The analysis shown will exclude cases in which there are more unknowns than equations but the developed procedure would work for either case.

Using the same notation as in Zurmühl [69,70], a set of linear and inhomogeneous equations with m unknowns can be combined into a single matrix equation of the form

$$A \bar{x} = \bar{a} \quad (53)$$

A is a coefficient matrix with n rows and m columns, \bar{x} represents a column vector of m elements and \bar{a} is a column vector of n elements. It is obvious that there is an exact solution to these equations only for the case in which \bar{a} represents a dependent vector to the column vectors in the matrix A . The probability of having such a situation becomes smaller with an increasing difference between m and n .

Assuming $n > m$ and using first a non-singular matrix A (the case of a singular matrix will be better understood after the demonstration of a non-singular case) with \bar{a} independent of A , the error can be represented by the vector \bar{e} with an equal number of elements as in \bar{a} . Equation (53) can be rewritten into the form

$$A \bar{x} - \bar{a} = \bar{e} \quad (54)$$

This equation will always have an exact solution because \bar{e} is simply calculated to suit the choice made for \bar{x} . The question now is how to find an \bar{x} which will minimize the elements of \bar{e} or the sum e .

$$\begin{aligned} e &= \sum_{i=1}^n e_i^2 = \bar{e}^T \cdot \bar{e} = (\bar{x}^T A^T - \bar{a}^T) \cdot (A \bar{x} - \bar{a}) \\ &= \bar{x}^T A^T A \bar{x} - 2 \bar{x}^T A^T \bar{a} + \bar{a}^T \bar{a} \end{aligned}$$

\bar{e}^T does represent the transposed vector of \bar{e} which is a row vector and

the expression $\bar{\mathbf{e}}^T \cdot \bar{\mathbf{e}}$ would be the same as the dot product. A^T represents the transposed matrix of A .

This last expression for e is a continuous function of \bar{x} and the minimum can be obtained by setting its derivative equal to zero.

$$\frac{de}{d\bar{x}} = 2 A^T A \bar{x} - 2 A^T \bar{a} = 0 \quad \text{or} \quad A^T A \bar{x} = A^T \bar{a} \quad (55)$$

The obtained form is apparently the requirement for the minimum in a least square sense, and it is possible to solve the equation for \bar{x}

$$\bar{x} = A^\dagger \bar{a} \quad \text{where} \quad A^\dagger = [A^T A]^{-1} A^T \quad (56)$$

The matrix A^\dagger which has m rows and n columns could be compared with an inverse of A and is called a "pseudoinverse." Using basic matrix calculus, it can be shown that, if A is non-singular and $n > m$, then also $A^T A$ is non-singular and is an $m \times m$ matrix which has a non-vanishing determinant and an inverse $[A^T A]^{-1}$. Therefore, A^\dagger does exist and yields a unique solution for \bar{x} . It will be shown later that the solution is unique only for a matrix with independent column vectors and that it does not matter whether or not it has any dependent row or column vectors. However, so far only a non-singular matrix should be considered with no dependency in the columns or in the rows.

If one considers equations (53) and (55) from a different viewpoint, it could be said that the multiplication with A^T represents the aim of transforming the system of equations in (53) into something that is non-linear and quadratic and, therefore, has a solution for \bar{x} . The resulting system should be reversible, or in other words, the matrix A must form an

orthogonal system to the pseudoinverse matrix A^\dagger such that

$$A^\dagger A = I \quad (57)$$

which can be proven by substituting for A^\dagger from equation (56).

It is also of some interest to investigate the case in which $n < m$ or where there are more unknowns than equations.

$$C \bar{x} = \bar{b} \quad (58)$$

Here the form $C^T C$ will become singular because the multiplication of the dependent column vectors will form again a dependent system of vectors. Therefore, no inversion of $C^T C$ does exist. However, the form $C C^T$ is found to be non-singular and can be inverted. But it is not possible to show a similar relation as in equation (57). If following a similar pattern, the solution for \bar{x} and the pseudoinverse would become

$$\bar{x} = C^\dagger \bar{b} \quad \text{where} \quad C^\dagger = C^T [C C^T]^{-1} \quad (59)$$

If one could prove the identity when substituting equation (59) into the left hand side of equation (55), then the found solution could be confirmed as a least square solution to the problem. It is possible to set

$$C^T C C^\dagger \bar{b} = C^T C C^T [C C^T]^{-1} \bar{b} = C^T \bar{b}$$

which proves the identity with equation (55). However, here equation (59) does not represent the only solution and the received result does represent only one of the possible values satisfying equation (58).

Because matrix C has fewer elements in its column vectors than it has columns, the column vectors must be dependent and, at best, there can be only as many independent column vectors as there are rows or ele-

ments in these vectors. If, for example, $m = n+1$, then the column vectors must have a relation

$$\alpha \cdot \bar{c}_1 + \beta \cdot \bar{c}_2 + \dots + \sigma \cdot \bar{c}_n = \bar{c}_{n+1} \quad (60)$$

in which the values $\alpha, \beta, \dots, \sigma$ are to be selected to suit the vectors \bar{c} . Now, using the column vectors $\bar{c}_1, \bar{c}_2, \dots, \bar{c}_{n+1}$ instead of the matrix C , equation (58) can be rewritten into the form

$$\bar{c}_1 \cdot x_1 + \bar{c}_2 \cdot x_2 + \dots + \bar{c}_n \cdot x_n + \bar{c}_{n+1} \cdot x_{n+1} = \bar{b}$$

Substituting equation (60) for \bar{c}_{n+1} will supply

$$\bar{c}_1 \cdot (x_1 + \alpha \cdot x_{n+1}) + \bar{c}_2 \cdot (x_2 + \beta \cdot x_{n+1}) + \dots + \bar{c}_n \cdot (x_n + \sigma \cdot x_{n+1}) = \bar{b} \quad (61)$$

which shows a new equation with only n variables shown in the parenthesis expressions as $\xi_1 = (x_1 + \alpha \cdot x_{n+1})$, $\xi_2 = (x_2 + \beta \cdot x_{n+1})$, \dots and $\xi_n = (x_n + \sigma \cdot x_{n+1})$. They are found by the solution to the $n \times n$ matrix of C equated to \bar{b} and have only one solution. If each new variable is solved for the original unknowns, it can be seen that they will allow an arbitrary choice for x_{n+1} and in this form permit an infinite number of solutions. The multitude becomes ∞^2 if m is larger than n by a factor of two. This conclusion is obtained not only for the case $n < m$ but also for cases in which the rank of the matrix A is smaller than m . The demonstration of the proof may not be the shortest one, but it can easily be modified to show the possible solutions for a square matrix with any type of dependency in the columns. It may be demonstrated by a similar equation, as shown in (60), and will allow a least square solution to a rectangular matrix similar to the square form of C in equation (61). And again, it

will be possible to find a multitude of best solutions.

Now the most general matrix inversion can be considered, which may include matrices with dependencies in both the columns and the rows. The standard inversion procedure developed for the square matrix will record the numbers of the rows and columns which are independent and will permit sorting out those vectors which cause the singularity. Then matrix A can be replaced by

$$A = B H C \quad \text{for } n > m \quad (62)$$

where B is an $n \times r$ non-singular matrix containing the independent columns of A, and C is also a non-singular but $r \times m$ matrix of all independent rows. H is a square matrix of the dimensions $r \times r$. Following equations (56) and (59), one can set

$$B^\dagger B = C C^\dagger = I$$

The relations

$$I_L = B B^\dagger \quad \text{and} \quad I_R = C^\dagger C$$

are called the left and the right identity matrix because, substituting for the pseudoinverse from equation (56)

$$I_L B = B B^\dagger B = B [B^T B]^{-1} B^T B = B$$

and from (59)

$$C I_R = C C^\dagger C = C C^T [C C^T]^{-1} C = C$$

And it is also possible to show that

$$I_L A = B [B^T B]^{-1} B^T B H C = B H C = A \quad (63)$$

$$A I_R = B H C C^T [C C^T]^{-1} C = B H C = A \quad (64)$$

Considering the identity of equation (62) as a product of three matrices which can be inverted by the least square method and should follow the same inversion rule as the non-singular square matrices, one should obtain for the pseudoinverse

$$A^\dagger = C^\dagger H^{-1} B^\dagger \quad (65)$$

Checking the form

$$A A^\dagger = B H C C^\dagger H^{-1} B^\dagger = B B^\dagger = I_L \quad (66)$$

because

$$C C^\dagger = H H^{-1} = I$$

and

$$A^\dagger A = C^\dagger H^{-1} B^\dagger B H C = C^\dagger C = I_R \quad (67)$$

It can be seen that the two forms agree with the previous equations (63) and (64) because

$$I_L A = B B^\dagger B H C = B H C = A$$

$$A I_R = B H C C^\dagger C = B H C = A$$

It should be noticed that none of the two forms (66) and (67) can become a unit matrix because of the singularities in the matrix A which do not exist in the matrix A of equation (57).

In order to apply this method effectively, one has to find a convenient way for locating H, or even better, H^{-1} . Using two new matrices J_L and J_R , which have only ones and zeros as elements and will allow the transformation of the matrix A into the matrices B and C by using the equations

$$B = A J_R \quad \text{and} \quad C = J_L A \quad (68)$$

Both J_R and J_L are constructed to bring only the independent columns of A into B and the independent rows of A into C. Premultiplying with C

and postmultiplying with B, equation (65) yields the form

$$C A^\dagger B = C C^\dagger H^{-1} B^\dagger B = H^{-1}$$

Substituting the forms of equation (68) into the above

$$H^{-1} = J_L A A^\dagger A J_R = J_L I_L A J_R = J_L A J_R$$

which indicates that the inverted matrix H can simply be found by taking all elements of the submatrix common to B and C.

And now it is possible to state the solution to the equation

$$A \bar{x} = \bar{a}$$

in the most general form by using the pseudoinverse $A^\dagger = C^\dagger H^{-1} B^\dagger$ in

$$\bar{x} = A^\dagger \bar{a}$$

The pseudoinverses of B and C are

$$B^\dagger = [B^T B]^{-1} B^T \quad \text{and} \quad C^\dagger = C^T [C C^T]^{-1}$$

The method outlined for the general inversion appears to be rather complicated but, in fact, comes out as a simple procedure. Using basic matrix calculus, one can see that the matrix $A^T A$ is symmetric and has the same dependency in the columns as the original matrix A. The matrix $A A^T$ has always a dependency because its rank is at the best equal to the number of columns m and all (n-m) vectors depend on the first m ones. However, if there was a row dependency amongst the first m rows, then this dependency will also be reproduced in the product matrix $A A^T$ and recognized by the matrix inversion program. When the inversion procedure halts because of the existing dependency, the inversion up to this part of the program becomes the matrix $[C C^T]^{-1}$. But before this is checked,

the first form $A^T A$ undergoes the test and will render either $[A^T A]^{-1}$ or for similar reasons $[B^T B]^{-1}$. When there is no dependency detected, then the form of

$$A^\dagger = [A^T A]^{-1} A^T$$

will be found. In case of a dependency, $[B^T B]^{-1}$ is stored for use in the general form of B^\dagger and the second test is performed rendering $C^\dagger = C^T [C C^T]^{-1}$. It is then not too difficult to find H^{-1} and to compute the finished form of

$$A^\dagger = C^\dagger H^{-1} B^\dagger$$

Here the diversity of the inversion program shows at its best and does make it possible to get by with only one or at the most two executions of the one-loop inversion procedure for the computation of A^\dagger in its most general form.

The Pseudoinverse Method with Exact Solutions

for a Selected Set of Equations

This procedure will complete the application of the inversion procedure together with the pseudoinverse and shall be added here to the presented material. First, it was thought to be of an advantage to the synthesis of the path-generator but did not contribute any additional convergency speed in the synthesis. Therefore, it was dropped from the program for a reason best understood later in the chapters on the gradient and the relaxation methods.

Dividing the matrix A into four submatrices such that

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$$

as proposed by Diederich [68], then the equation $A \bar{x} = \bar{a}$ can be divided into two matrix equations

$$A_1 \bar{x}_1 + A_2 \bar{x}_2 = \bar{a}_1 \quad (69)$$

$$A_3 \bar{x}_1 + A_4 \bar{x}_2 = \bar{a}_2 \quad (70)$$

A_1 , A_2 , and \bar{a}_1 do not need to be composed of the first few rows. They simply contain those rows which represent equations to be kept exact and the remaining rows or equations are contained in A_3 , A_4 , and \bar{a}_2 . This means that the vector \bar{a} is divided in an equal manner into two subvectors \bar{a}_1 and \bar{a}_2 . However, A_1 must be a non-singular square matrix with as many columns as it has rows. The columns are selected by a dependency test in the standard matrix inversion by inverting the form $A_1^* A_1^{*T}$ to $[A_1^* A_1^{*T}]^{-1}$. In this stage, A_1^* will contain all row elements of A_1 and A_2 . After this the independent elements are picked for A_1 to assure an equal amount of independent columns to the selected rows and the remainder is collected in the matrix A_2 . Also the variable vector \bar{x} is divided equally into two subvectors \bar{x}_1 and \bar{x}_2 . The matrices A_3 and A_4 contain the remaining elements not included in the first two but have the same columns as the ones used for A_1 or A_2 , respectively.

When the four submatrices and subvectors are selected in this manner, it is not too difficult to solve the two matrix equations (69) and (70) for the unknown vectors \bar{x}_1 and \bar{x}_2 . Solving (69) for \bar{x}_1 , it becomes

$$\bar{x}_1 = A_1^{-1} (\bar{a}_1 - A_2 \bar{x}_2) \quad (71)$$

which, if substituted into equation (70), will render a solution for \bar{x}_2 .

$$A_3 A_1^{-1} (\bar{a}_1 - A_2 \bar{x}_2) + A_4 \bar{x}_2 = \bar{a}_2$$

or
$$Q \bar{x}_2 = \bar{q} \quad \text{and} \quad \bar{x}_2 = Q^\dagger \bar{q}$$

where
$$Q = A_4 - A_3 A_1^{-1} A_2$$

and
$$\bar{q} = \bar{a}_2 - A_3 A_1^{-1} \bar{a}_1$$

After the computation of \bar{x}_2 , it is not difficult to substitute these values into equation (71) to find \bar{x}_1 . This method will require only one extra matrix inversion for A_1^{-1} and one pseudoinverse for Q^\dagger . The rest consists of ordinary matrix multiplications and summations which can be computerized without difficulties.

CHAPTER VI.

DIFFERENTIATION METHOD

General Comments

The synthesis method presented is based on actual error measurements taken from trial runs of the analog simulation which had to be used as substitute for the performance of a hybrid computer. It was not possible to use analytical expressions because of their complexity and because it was impossible to find the required explicit forms. Under those conditions one can consider the entire computer program as a function which does produce the desired sum e of all measured errors ϵ for a given set of physical dimensions of the four-bar mechanism. There are nine such dimensions or parameters which specify a mechanism and which have to be fed to the program to obtain the desired result for e . These parameters are independent from each other and can be considered as elements of a nine dimensional vector. In such a way the sum of all errors e would form a function of the type

$$e = f(\bar{p}) \quad (72)$$

The symbol f could stand for any type of function but in this case stands for the part of the computer program which does simulate the motion of the mechanism and takes an account of each error ϵ between the obtained coupler point position and the actually desired position at each listed precision point. (See chapter IV for error definition and analog simulation.)

This condition makes it impossible to use a standard differentiation method for the function f . The trend found most often is to use an approximation based on a polynomial substitution for the function f . For example, the expression

$$e_i = a_0 + a_1 p_i + a_2 p_i^2 + \dots + a_r p_i^r = \bar{p}_i \bar{a}$$

would represent such a polynomial. The vector \bar{p}_i has the components $1, p_i, p_i^2, \dots, p_i^r$ and the constants $a_0, a_1, a_2, \dots, a_r$ would have to be found by calculating e_i for various values of \bar{p}_i . When using the matrix relation the required equations are

$$\bar{e} = [P] \bar{a} \quad \text{or} \quad \bar{a} = [P]^{-1} \bar{e}$$

For a given matrix P and a computed \bar{e} , the constants \bar{a} can be found. There are $r+1$ equations needed for computing all $r+1$ coefficients of \bar{a} . The row elements in P could have the form $1, p, p^2, p^3, \dots, p^r$ but they also could have an increasing negative power. Any form is useful as long as the components represent a linear independent set of polynomials.

For a better accuracy, a larger number of a 's could be needed, in spite of a skillfully selected set of polynomials in P . Each equation in P will require a complete run of the analog simulation and, therefore, becomes a time consuming item in the computer program. There is even one additional run needed for a supplementary coefficient in \bar{a} which is generally used to determine the accuracy of the polynomial approximation. If the value of \bar{e} does not considerably change by adding this one more term, then the polynomial will not need to be expanded any further and the differentiation of the obtained function can be performed.

In order to eliminate this time consuming cycling of the analog simulation, an effort was made to replace this method by a completely different type of analysis for obtaining the derivative of the function in equation (72).

Standard Estimate of Differentiation Accuracy

In order to comply with the standard symbols for the dependent and independent variables, equation (72) is changed to the form

$$y = y(\bar{x}) \quad (73)$$

in which the vector \bar{x} could represent the nine independent parameters of the mechanism system and y the objective variable which should be minimized. The method at hand would be to alter \bar{x} by an increment $\Delta\bar{x}$ and to check the effect on y . This could be expressed best by the Taylor expansion

$$y_2 = y_1 + \Delta\bar{x} \cdot \frac{\partial y_1}{\partial \bar{x}} + \frac{(\Delta\bar{x} \cdot \nabla)^2}{2!} (y_1) + \frac{(\Delta\bar{x} \cdot \nabla)^3}{3!} (y_1) + \dots \quad (74)$$

This series remains bounded for a small $\Delta\bar{x}$ and $\frac{\partial y_1}{\partial \bar{x}}$ actually represents the gradient at the point y_1 . Because of the fact that no differentiation can be executed, the value for $\frac{\partial y_1}{\partial \bar{x}}$ will have to be approximated by an expression

$$\frac{\partial y_1}{\partial \bar{x}} = \frac{y_2 - y_1}{\Delta\bar{x}} - \frac{(\Delta\bar{x} \cdot \nabla)^2}{2! \Delta\bar{x}} (y_1) - \dots \quad (75)$$

This gradient vector can be broken down into the individual components of $\Delta\bar{x}$ and then each partial differential would have the form

$$\frac{\partial y_1}{\partial x} = \frac{y_2 - y_1}{\Delta x} - \frac{\Delta x}{2!} \frac{\partial^2 y_1}{\partial x^2} - \frac{\Delta x^2}{3!} \frac{\partial^3 y_1}{\partial x^3} - \dots \quad (76)$$

If one would plot the curve $y = y(x)$ in a plane coordinate system, then the factor $\frac{\partial y_1}{\partial x}$ becomes $\frac{dy_1}{dx}$ and is the direction of the tangent at the point y_1 , whereas the expression $\frac{y_2 - y_1}{\Delta x}$ is the direction of the secant between the points y_1 and y_2 , Figure 21. It can be seen that the slope of the secant and the slope of the tangent are not the same and the slope difference is expressed in equation (76) by the higher order terms which may be a negligible amount for flatter curves or for a smaller Δx . Because there is no control over the shape of the curve, the problem becomes to select the value of Δx for a desired accuracy of $\frac{dy_1}{dx}$ or in our case of $\frac{\partial y_1}{\partial x}$.

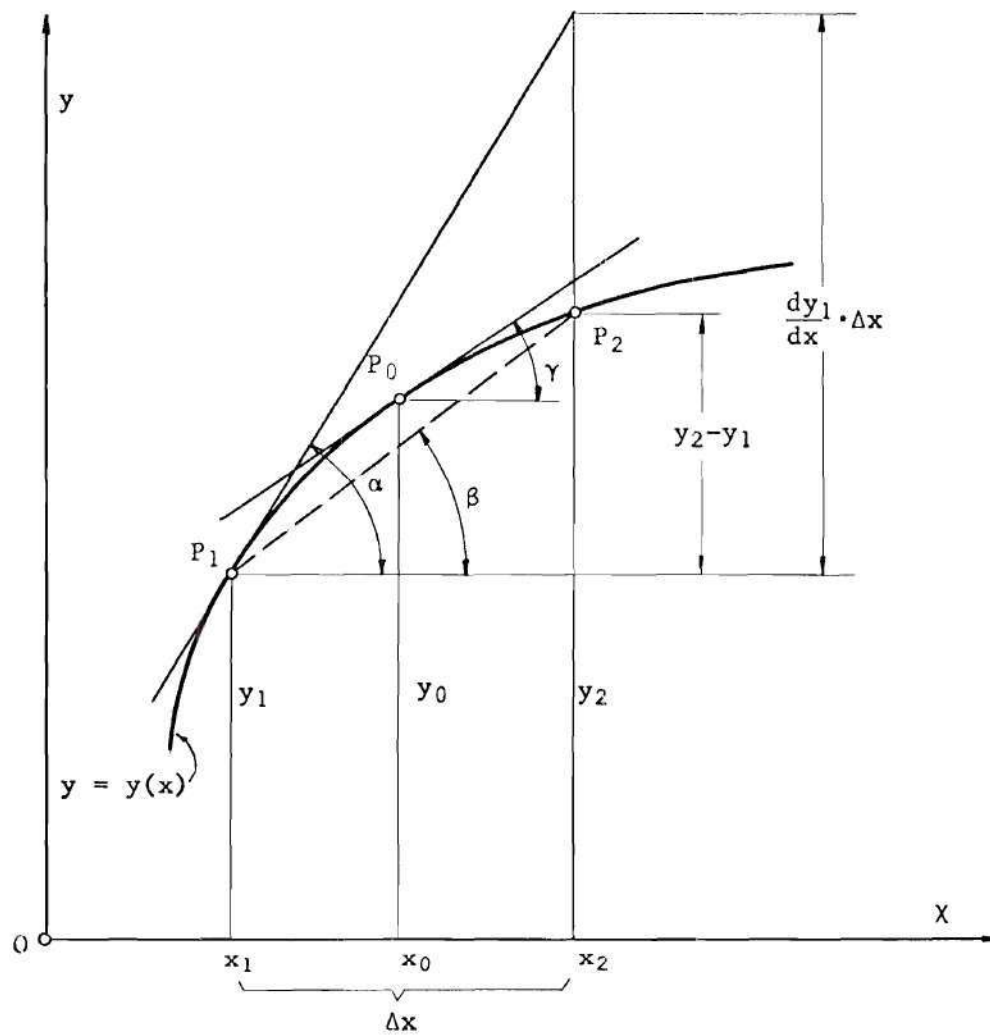
The first step of improvement would be to select a third point for the estimated differential. Figure 21 shows quite clearly that, for a reasonably small Δx , the approximating secant would replace much better a tangent at a midpoint x_0 between x_1 and x_2 . Such a change is introduced by adding the point x_0 with a distance of $-\frac{\Delta x}{2}$ to x_1 and a distance of $+\frac{\Delta x}{2}$ to x_2 . For this arrangement, the partial differential will become

$$\frac{\partial y_0}{\partial x} = \frac{y_2 - y_1}{\Delta x} - (\text{higher order terms})$$

The accuracy of this expression can be shown again by the Taylor expansions

$$y_1 = y_0 - \frac{\Delta x}{2} \frac{\partial y_0}{\partial x} + \frac{\Delta x^2}{2^2 2!} \frac{\partial^2 y_0}{\partial x^2} - \frac{\Delta x^3}{2^3 3!} \frac{\partial^3 y_0}{\partial x^3} + \dots$$

$$y_2 = y_0 + \frac{\Delta x}{2} \frac{\partial y_0}{\partial x} + \frac{\Delta x^2}{2^2 2!} \frac{\partial^2 y_0}{\partial x^2} + \frac{\Delta x^3}{2^3 3!} \frac{\partial^3 y_0}{\partial x^3} + \dots$$



$$\operatorname{tg} \alpha = \frac{dy_1}{dx}$$

$$\operatorname{tg} \beta = \frac{y_2 - y_1}{\Delta x}$$

$$\operatorname{tg} \gamma = \frac{dy_0}{dx}$$

Figure 21. Various Differentiation Methods

And the difference divided by Δx becomes

$$\frac{\partial y_0}{\partial x} = \frac{y_2 - y_1}{\Delta x} - \frac{\Delta x^2}{2^2 3!} \frac{\partial^3 y_0}{\partial x^3} - \frac{\Delta x^4}{2^4 5!} \frac{\partial^5 y_0}{\partial x^5} - \dots \quad (77)$$

which shows quite clearly an improvement against the form of equation (76). There the error started with a term $\frac{\Delta x}{2!} \frac{\partial^2 y_1}{\partial x^2}$ whereas in the new form it starts with the term $\frac{\Delta x^2}{2^2 3!} \frac{\partial^3 y_0}{\partial x^3}$. This proves the findings from the diagram in Figure 21, but it still does not show what the error is from neglecting all higher terms in equation (77). For estimating these terms, one would need to calculate the third and possibly fifth derivatives which would require the addition of further points on our curve $y = y(x)$ and again would cause additional computer time. The three points, as needed for the proposed method of the first derivative, would permit the approximation of the second derivative, but for the higher derivatives further points are needed. It is for this reason that the proposed circle point method, shown next, became so useful, because it requires only three points and still allows an estimate for its accuracy.

Circle Point Method

The described method will be most effective for smooth curves but may also be applied within limitations to other curves. However, it will be a complete failure for cusps with only one tangent direction or a very small angle between the tangent lines. A smooth curve should be understood mathematically as a curve with a continuous derivative. As it is known, it is possible for the coupler curve of a four-bar mechanism to have as many as three cusps and in general cannot be considered, for that

rare case, as a smooth curve. The function for e of equation (72) is shown in more detail in the paragraph on Error Definition in chapter IV which indicates that it includes the coupler curve coordinates and their derivatives. The circle point method proved effective with the exception of the Joukowski profile problem which ordinarily has one cusp. Here, some logical expressions had to be built into the program in addition to the developed circle point method.

Coming back to the representation of equation (73) and considering only one variable of the vector \bar{x} , the partial derivative can be seen as a simple plane problem and could be worded as follows:

A function $y = y(x)$ shall be given by three closely spaced points. Find the derivative at the midpoint (x_0, y_0) and space the points to suit the required accuracy.

If a smooth curve has three closely spaced points, then it should always be possible to put an approximation circle through those points. The derivative of such a circle equation should then become a good approximation for the desired derivative of the original curve, that is provided it was possible to fit the circle very close to the desired curve area. The checking of this feature will be done later, but first the computation of the circle equation and its derivative shall be discussed.

A. Derivative of the Approximation Circle

Figure 22 shows an approximation circle to a curve related to a cartesian coordinate system with its y-axis through the point P_0 . This special orientation becomes imperative for making the calculation less sophisticated but has no effect on the general end result. The derivative $\frac{dy}{dx}$ does not depend on the location of a coordinate system as long

as the angular direction is maintained. For the coordinate system (ξ, η) with its origin in P_0 , the equation of the approximation circle becomes

$$(\xi-a)^2 + (\eta-b)^2 - r^2 = 0 \quad (78)$$

where $a^2 + b^2 = r^2$. The differentiation of this equation gives the values for the sought derivative.

$$2(\xi-a) d\xi + 2(\eta-b) d\eta = 0 \quad \text{and} \quad \frac{dy}{dx} = \frac{d\eta}{d\xi} = \frac{\xi-a}{b-\eta} \quad (79)$$

The general equation for the approximation circle through three given points can be found by the evaluation of the determinant

$$\begin{vmatrix} \xi^2 + \eta^2 & \xi & \eta & 1 \\ \xi_1^2 + \eta_1^2 & \xi_1 & \eta_1 & 1 \\ 0 & 0 & 0 & 1 \\ \xi_2^2 + \eta_2^2 & \xi_2 & \eta_2 & 1 \end{vmatrix} = 0$$

or what is the same

$$\begin{aligned} & (\xi^2 + \eta^2)(\xi_1 \eta_2 - \xi_2 \eta_1) - \xi [(\xi_1^2 + \eta_1^2)\eta_2 - (\xi_2^2 + \eta_2^2)\eta_1] \\ & + \eta [(\xi_1^2 + \eta_1^2)\xi_2 - (\xi_2^2 + \eta_2^2)\xi_1] = 0 \end{aligned}$$

The second point has the coordinates $(0,0)$ and the first and third points have the coordinates carrying the subscripts 1 and 2. The values for a and b of equations (78) and (79) can now be calculated as

$$2 a = \frac{(\xi_1^2 + \eta_1^2) \eta_2 - (\xi_2^2 + \eta_2^2) \eta_1}{\xi_1 \eta_2 - \xi_2 \eta_1}$$

$$- 2 b = \frac{(\xi_1^2 + \eta_1^2) \xi_2 - (\xi_2^2 + \eta_2^2) \xi_1}{\xi_1 \eta_2 - \xi_2 \eta_1}$$

The derivative of equation (79) evaluated for the point $(\xi_0=0, \eta_0=0)$ becomes

$$\frac{d\eta_0}{d\xi} = - \frac{a}{b}$$

and substituting the values for a and b

$$\frac{d\eta_0}{d\xi} = \frac{(\xi_1^2 + \eta_1^2) \eta_2 - (\xi_2^2 + \eta_2^2) \eta_1}{(\xi_1^2 + \eta_1^2) \xi_2 - (\xi_2^2 + \eta_2^2) \xi_1} = \frac{\eta_1 \eta_2 (\eta_1 - \eta_2) + \xi_1^2 \eta_2 - \xi_2^2 \eta_1}{\xi_1 \xi_2 (\xi_1 - \xi_2) + \eta_1^2 \xi_2 - \eta_2^2 \xi_1}$$

Finally, it is necessary to transfer the coordinates back to the general coordinates x, y and to substitute the quantities

$$\xi_1 = -\frac{\Delta x}{2}, \quad \eta_1 = y_1 - y_0, \quad d\xi = dx$$

$$\xi_2 = \frac{\Delta x}{2}, \quad \eta_2 = y_2 - y_0, \quad d\eta = dy$$

This will bring the derivative for the point $(0, y_0)$, as shown in Figure 22, to the form

$$\begin{aligned} \frac{dy_0}{dx} &= \frac{2 (y_1 - y_0)(y_2 - y_0)(y_1 - y_2) + \Delta x^2/2 (y_2 - y_1)}{\Delta x^3/2 + \Delta x [(y_0 - y_1)^2 + (y_2 - y_0)^2]} \\ &= \frac{(y_2 - y_1)}{\Delta x} \cdot \frac{[\Delta x^2/2 + 2 (y_0 - y_1)(y_2 - y_0)]}{[\Delta x^2/2 + (y_0 - y_1)^2 + (y_2 - y_0)^2]} \end{aligned} \quad (80)$$

Comparing this equation with the expression of equation (77), one finds that the factors of higher orders are now replaced by a rather simple multiplying factor and can be shown as

$$\begin{aligned} & \frac{\Delta x^2}{2^2 3!} \frac{\partial^3 y_0}{\partial x^3} + \frac{\Delta x^4}{2^4 5!} \frac{\partial^5 y_0}{\partial x^5} + \dots \\ &= \frac{(y_2 - y_1)}{\Delta x} \left\{ \frac{\Delta x^2/2 + 2(y_0 - y_1)(y_2 - y_0)}{\Delta x^2/2 + (y_0 - y_1)^2 + (y_2 - y_0)^2} \right\} \end{aligned}$$

because the partial derivative of equation (77) actually has the same meaning as the total derivative of equation (80). The replacement of the infinite sum by a finite product becomes a definite advantage provided that the approximating circle does represent an exact replica of the curve at the point y_0 and its immediate surrounding. This condition will depend entirely on the selection of the value for the increment Δx , and it shall be shown next how to check this selected value for Δx .

B. Selection of the Finite Increment for the Desired Accuracy

Figure 23 shows the approximation circle with its three points P_1 , P_0 , and P_2 on the actual curve. It is obvious that the circle will approximate the curve at the point P_0 best when the three points are infinitely close together. This means that also the angle between the two secants will become zero, or $\beta - \alpha \rightarrow 0$ for $\frac{\Delta x}{2} \rightarrow 0$. That is, a certain conformity of the circle to the curve will require smaller increments for smaller radii, but the angle difference $\beta - \alpha$ will remain the same for any radius r . That means that a flatter portion of the curve with a larger radius will have a good approximation with larger values for Δx whereas a portion with a smaller r can give only a poor approximation if the

value Δx is not considerably reduced. However, this argument cannot be applied to the angle between the two secants at the point P_0 . This angle will always indicate the quality of the approximation circle whether it has a larger or a smaller radius. Selecting its value close to zero will, therefore, always render a good approximation.

The experience showed that a value of 8 to 9 degrees was satisfactory. The computer program will first select a small estimated value for Δx and then check the obtained angle. Should it become too large, the program will reduce Δx by an appropriate amount and renew the checking process.

It is not very complicated to calculate the angles α and β using the trigonometric relations

$$\operatorname{tg}(\beta - \alpha) = \frac{\operatorname{tg}\beta - \operatorname{tg}\alpha}{1 + \operatorname{tg}\beta \cdot \operatorname{tg}\alpha} \quad \text{or} \quad \operatorname{tg}|\beta - \alpha| = \frac{|\operatorname{tg}\beta - \operatorname{tg}\alpha|}{1 + \operatorname{tg}\beta \cdot \operatorname{tg}\alpha}$$

and because of the needed relation $\delta > |\beta - \alpha|$ it follows

$$\operatorname{tg}\delta > \frac{|\operatorname{tg}\beta - \operatorname{tg}\alpha|}{1 + \operatorname{tg}\beta \cdot \operatorname{tg}\alpha} \quad \text{and} \quad 1 + \operatorname{tg}\beta \cdot \operatorname{tg}\alpha > \frac{|\operatorname{tg}\beta - \operatorname{tg}\alpha|}{\operatorname{tg}\delta}$$

Dealing only with tangent functions, one can substitute for them the values

$$\operatorname{tg}\alpha = 2 \frac{y_0 - y_1}{\Delta x} \quad \text{and} \quad \operatorname{tg}\beta = 2 \frac{y_2 - y_0}{\Delta x}$$

and the following inequality can be obtained

$$1 + \frac{4}{\Delta x^2} (y_0 - y_1)(y_2 - y_0) > \frac{1}{\text{tg}^2} |(y_2 - y_0) - (y_0 - y_1)| \cdot \frac{2}{\Delta x}$$

or

$$\frac{\Delta x^2}{2} + 2(y_0 - y_1)(y_2 - y_0) > \frac{\Delta x}{\text{tg}^2} |(y_2 - y_0) - (y_0 - y_1)| \quad (81)$$

This relation turns out to be very handy because it contains the numerator of equation (80), which needs only to be compared with the right hand side of equation (81) to determine whether one has made an adequate choice for Δx or not.

If one would make the inequality (81) an equality for an attempt to find the needed minimum Δx , one would receive a very poor result, because it should not be overlooked that also the two values of y_1 and y_2 are dependent on the choice of Δx . Any other attempts for a better estimate of Δx , if it were found not to satisfy inequality (81), involve quadratic equations, and it appears that a mere second guess is just as speedy and effective.

CHAPTER VII

GRADIENT METHOD

General Comments

Essentially there are two types of gradient method which could be applied to the problem; their advantages and disadvantages will be explained in this chapter. Many researchers [44,66,71] who have applied the gradient method to this problem have discovered that it was not a foolproof method for problems with hard-to-guess initial approximations. And further, it was discovered that this method will not converge at all, for cases in which the initial stages are too far off the sought end result. This behavior will be explained later. However, one cannot deny the good results which eventuate when this method is used with a good initial guess or for only the final steps of any given problem.

It was found that there is no need for introducing the limiting conditions which are mentioned as added features in numerous other publications, because the problem setting of this investigation is considerably different from most others. Here the question is whether or not there is a four-bar mechanism which is able to synthesize best the given path with no concern given to the mechanical features of such mechanisms. When implying certain restrictions to the physical dimensions of the mechanism, one is risking the elimination of the only good solution to a given path problem. If, with the solution at hand, it is felt that it does not meet with the physical or dynamic requirements, one could still investigate the two cognate solutions which are different in their dimen-

sions but, as one knows, will trace exactly the same coupler curve.

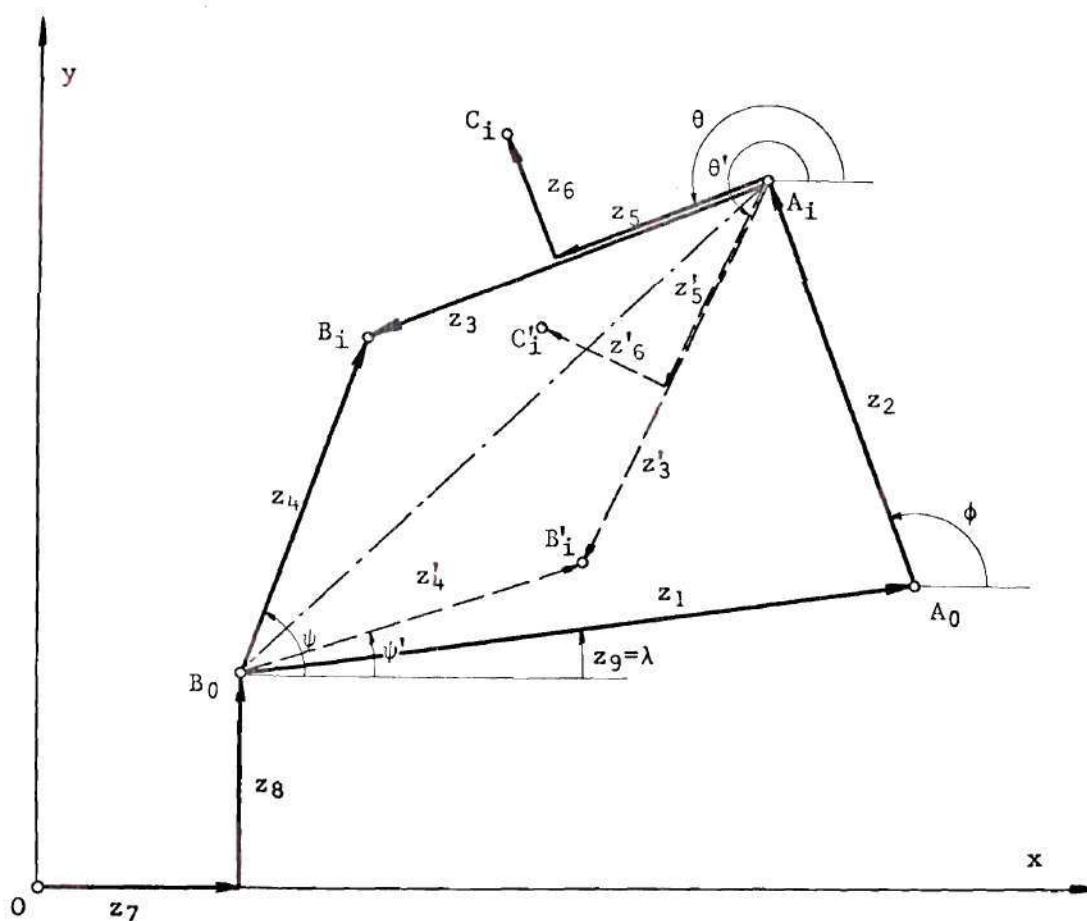
Hence, the problem setting in this paper is not concerned with the applicability of any solution but only with what is the solution with a minimum accumulated error. However, it is important to observe that none of the dimensions of the links forming the four-bar ring could become negative. This, in a sense, represents a restriction and at the beginning it was felt that it might be necessary to introduce an expression which would safeguard the computations against such a trend. But the experience proved that the small steps of each iteration which might have such a prevailing trend would lead sooner or later to very small dimensions of one link. This would produce a ring of a very limited mobility and would for this reason ordinarily not at all improve the fitting of the generated path, and because each iteration will represent an improved approximation to the desired path, one does not need to fear that one of the link dimensions could become too small or even worse, turn out to be negative.

It is hard to demonstrate that here one actually has to deal with a nine dimensional space which is basically divided into the regions of the three cognates. It was interesting to observe that, once the approximation got under way in the area of one cognate, it never would result in another area and will stay within its bounds. It may well be that the reason for this behavior can be found in the fact that for switching from one area to another one of the link dimensions inside the mechanism ring, it would have to change from positive to negative, or what would represent the same, turn by 180 degrees without changing the positive sign. This, together with what was said earlier, should be able to ex-

plain why the approximation appears to divide the hyperspace into three parts. Future research efforts may well be able to show some relations between the facts of having three cognate mechanisms for a system which combines four links and the changing sign of links inside the mechanism ring. This might bring a new approach to the existence of cognates as shown by Roberts and Cayley [9,10].

Hyperspace of Four-Bar Mechanism

As it was shown earlier, there are nine parameters or informations needed to specify completely all dimensions of a four-bar mechanism. Figure 24 shows these dimensions as they are used in the computer program. There are eight lengths and one angle (λ or z_9) which indicate the angular orientation of the mechanism frame z_1 . The angles ϕ , θ , and ψ are not needed for the specification of the mechanism but indicate the momentary position of the links z_2 , z_3 , up to z_8 . The nine parameters are fully independent from each other whereas the three angles ϕ , θ , and ψ are not. Either one of the angles would specify the momentary position of all links. If the angles ϕ or ψ are used, the positions would be equivocal for a possible symmetry position of the coupler link. If, for example, the angle ϕ is specified, it would leave two alternate positions for the angles θ and ψ . But once a selection between the two possibilities θ and θ' is made, the last angle ψ becomes defined unequivocally. Figure 24 shows the two alternate positions of the coupler point C marked C and C' for a selected angle ϕ . It is not too difficult to see that both positions of C are on the same coupler curve if $z_1 + z_2 \geq z_3 + z_4$ that is the case in which link z_2 has a limited motion. If, however, $z_1 + z_2 < z_3 + z_4$ then it is found that the two positions of C do belong to



z_1 . . . frame	A_0, B_0 . fixed pins
z_2 . . . crank	A_i, B_i . moving pins
z_3 . . . coupler	C_i, C'_i . coupler points
z_4 . . . follower	ϕ . . . crank angle
z_5, z_6 . coupler point position in moving plane	θ, θ' . . coupler angles
z_7, z_8 . mechanism position in fixed plane	ψ, ψ' . . follower angles
z_9 . . . angular position of mechanism	λ . . . position angle

Figure 24. Z-Components of Nine Dimensional Hyperspace

two different curves which have no connections with each other and where the second position cannot be reached once the mechanism is assembled and set in motion on the first of the two alternate positions.

For the actual synthesis, only these nine parameters are important. Being independent from each other, they could be seen as filling a space of nine dimensions. Each point of this space would specify a certain mechanism with two different assemblies of the coupler link as shown above. We know that the first four coordinates of this hyperspace can only be positive, but the remaining ones could also become negative.

Let us stop here for a moment and evaluate all the possible mechanisms in this space. Suppose that we had to consider only a three dimensional space with only positive coordinates. The number of all possible points in such a space would depend on a grid which is applied to the count of variations and on the volume or the maximum dimensions allowed. It has no meaning to permit any link to become infinite long if one has no intention of including crank slider motions. Suppose the maximum length is f and the fineness of the grid is counted in q points per one unit of f . Then this space would be filled by $(fq)^3$ points. Coming back to the nine dimensional problem and again considering only positive dimensions, the number of points has increased to $(fq)^9$. If one adds the possible negative dimensions of the coordinates z_5 , z_6 , z_7 , and z_8 (the negative angle λ could be replaced by a respective positive supplementary angle), the result becomes

$$(fq)^6 \times (2fq)^4 = 2^4 \times (fq)^9$$

This indicates a rather large number, even for this computer age! The

reciprocal of this number gives an estimate of the probability to receive at random a particular linkage system out of the possible combinations and makes it hard to understand that any of the proposed random solutions will yield answers in a reasonable time [71,72]. It might help to find an initial mechanism to which a convergent minimization procedure could be applied. However, if there is a well performing minimization process, then almost any estimate for the initial mechanism would be sufficient.

Any point of the nine dimensional hyperspace will carry a certain value for the total error

$$e = \sum_{i=1}^n |\epsilon_i| \quad \text{or} \quad e = \sum_{i=1}^n \epsilon_i^2 \quad (82)$$

where ϵ_i is the error at the desired i^{th} precision point on the given path. All points with equal values for e are on a surface and the gradient has the direction normal to this surface. The gradient vector would present itself by the vector equation

$$\bar{g} = \frac{\partial e}{\partial z_1} \hat{i}_1 + \frac{\partial e}{\partial z_2} \hat{i}_2 + \frac{\partial e}{\partial z_3} \hat{i}_3 + \dots + \frac{\partial e}{\partial z_9} \hat{i}_9 \quad (83)$$

in which $z_1, z_2, z_3, \dots, z_9$ are the nine independent parameter coordinates. The vector \bar{g} is pointing in the direction with a maximum change of the value for e . However, is there really such a smooth surface to which the gradient has a unique direction? The summation of the absolute values in equation (82) will not have a continuous derivative and, for this reason, the second equation with the summation of the squares must

be applied.

But there is also another reason why equation (83) does not become too promising even when a summation of the squares is used. The needed derivatives for the vector \bar{g} will be continuous but cannot give any account for a sign change in the error elements ϵ_i . This can lead to misrepresentations and could bring about a wrong component for the gradient vector.

To understand this better, the diagram of Figure 25 was drawn and shows the errors ϵ_i in relation to the coupler curve length. To simplify matters, it is assumed that the distance s between the coupler point positions required for each precision point is a constant unit and can be stretched out along the abscissa of the shown diagram. This will make the area under the diagram curve ϵ_i approximately equal to the value

$$e_1 = \sum_{i=1}^n \epsilon_i (\bar{z} - \Delta z_r / 2)$$

The approximation will become better if more precision points are added in between the unit spaces of s , or what is the same, if n approaches infinity in the above sum. Changing one of the mechanism parameters z_r for an amount of Δz_r , a second such curve ϵ_2 is obtained and the area can be represented by a sum

$$e_2 = \sum_{i=1}^n \epsilon_i (\bar{z} + \Delta z_r / 2)$$

This change may also have some effect on the total length of the coupler curve and was indicated in the shown diagram. Now, squaring the ordinates

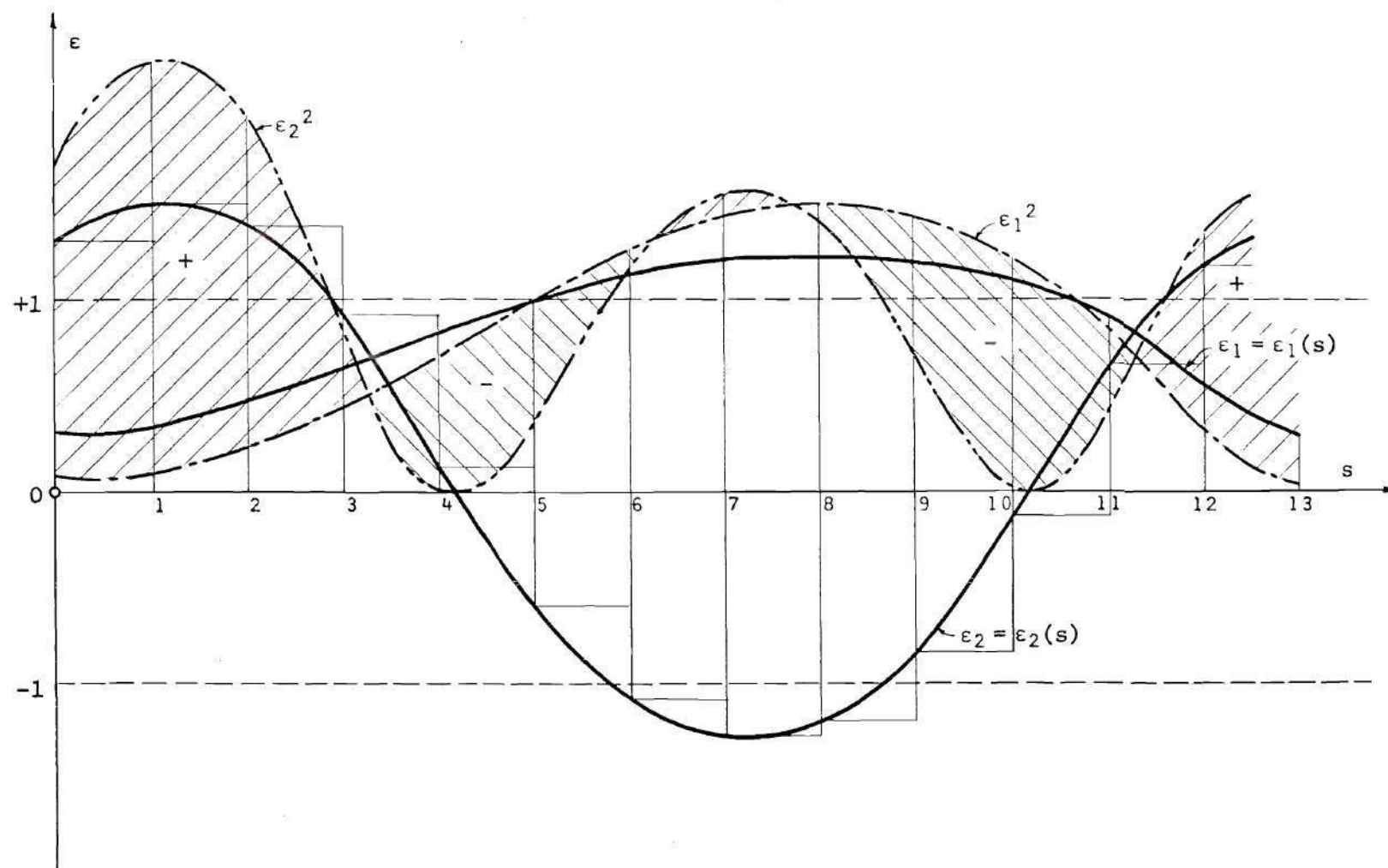


Figure 25. Analysis of Gradient Vector

of both curves will generate some new areas with the meaning

$$e_1 = \sum_{i=1}^n [\epsilon_i (\bar{z} - \Delta z_r / 2)]^2 \quad \text{and} \quad e_2 = \sum_{i=1}^n [\epsilon_i (\bar{z} + \Delta z_r / 2)]^2$$

The r^{th} gradient component of \bar{g} will become

$$\frac{\partial e}{\partial z_r} = \frac{e_2 - e_1}{\Delta z_r}$$

and can be found by dividing the area difference of the diagram in Figure 25 by Δz_r . A correction factor, as shown in equation (80), should be applied but is not essential for the intended consideration.

Now, it is not too difficult to see that the actual effect from the parameter change Δz_r was largest between the points 5 and 10, whereas the area after the squaring between the same points actually becomes smallest. There are positive and negative area sections, but even then, when all sections are taken as absolute values, the total area will never be able to represent the effect from the change of Δz_r and will, therefore, render an erroneous gradient component. It is true that the squaring process will minimize small values and increase larger ones and, in this fashion, help to correct the larger errors first, but as the example shows, the cancellation of the signs will upset considerably the represented effectiveness of any parameter change. It is for this reason that this gradient was not used and a different method, which considers much better each individual precision point, was used.

Definition of the Ninefold Optimization Vector $\Delta \bar{h}$

As it was shown in the previous paragraph, the minimization of the value e is hampered by a process which does exclude the signs of each individual ϵ_i . Only when these signs can be included for the computation of the gradient vector \bar{g} will the result be a true representative of the actual effects from any parameter changes. This cognition required a different type of gradient which was found in a combination of gradient vectors for each precision point. Any one of those individual gradients will point in the direction of the largest change at the precision point for which it was defined. All these gradients together will form a cluster of vectors of which a new direction $\Delta \bar{h}$ has to be found which will satisfy best the required changes at each precision point.

Because of the analog simulation, it was not difficult to establish a sign convention for the individual values of ϵ_i and in this form a record was made for whether or not a particular parameter change would go beyond the zero error line. This, together with the developed circle point differentiation method, made it possible to calculate the partial differentials of ϵ_i for the i^{th} precision point and became from equation (80)

$$\frac{\partial \epsilon_i}{\partial z_r} = \frac{\epsilon_{i2} - \epsilon_{i1}}{\Delta z_r} \frac{\Delta z_r^2 / 2 + 2(\epsilon_{i0} + \epsilon_{i1})(\epsilon_{i2} - \epsilon_{i0})}{\Delta z_r^2 / 2 + (\epsilon_{i0} - \epsilon_{i1})^2 + (\epsilon_{i2} - \epsilon_{i0})^2}$$

After the check from inequality (81) is made for each partial derivative, the gradient for the i^{th} precision point will read

$$\bar{e}_i + \frac{\partial \epsilon_i}{\partial z_1} \cdot \hat{i}_1 + \frac{\partial \epsilon_i}{\partial z_2} \cdot \hat{i}_2 + \dots + \frac{\partial \epsilon_i}{\partial z_9} \cdot \hat{i}_9 \quad (84)$$

Comparing this expression with equation (83), it is evident that the values of ϵ_i are differentiated instead of e and in this manner the signs of the error corrections are maintained.

In order to eliminate ϵ_i , that is the error at the i^{th} precision point, the parameters \bar{z} must be changed by an amount $\Delta \bar{h}$. Following the Taylor relation

$$\epsilon_i^* = \epsilon_i + \Delta \bar{h} \cdot \frac{\partial \epsilon_i}{\partial \bar{z}} + \frac{(\Delta \bar{h} \cdot \nabla)^2}{2!} (\epsilon_i) + \frac{(\Delta \bar{h} \cdot \nabla)^3}{3!} (\epsilon_i) + \dots \quad (85)$$

it can be seen that the first term $\frac{\partial \epsilon_i}{\partial \bar{z}}$ represents the gradient vector and the remaining terms can become negligible for a small $\Delta \bar{h}$. This indicates that, for larger values of ϵ_i , the value ϵ_i^* cannot become zero because $\Delta \bar{h}$ has to be kept small. It is for the high nonlinearity of the involved function that only very small steps can be made and only a very slow progress in reducing ϵ_i^* can be expected. The equation shows clearly that the higher order terms can only be neglected with small values for $\Delta \bar{h}$. Therefore, the gradient vector can only be used for computing $\Delta \bar{h}$ when ϵ_i is already reduced by other methods to a very small quantity.

Each precision point will have such a gradient which can be put together as the row vectors of the matrix G. This matrix will have as many rows as there are precision points and will have nine columns, the number of independent parameters of our four-bar mechanism. If now, all ϵ_i became small and, therefore, the required $\Delta \bar{h}$ would also be reasonably

small, one could eliminate all higher order terms in equation (85) and set for the i^{th} precision point

$$\epsilon_i^* = \epsilon_i + \Delta \bar{h} \cdot \bar{g}_i$$

Introducing now the fact that all ϵ_i are assumed to be small and that ϵ_i^* could be reduced to zero, the relation for all precision points is obtained by the matrix equations

$$\bar{\epsilon} = -G \cdot \Delta \bar{h} \quad \text{and} \quad \Delta \bar{h} = -G^{\dagger} \cdot \bar{\epsilon} \quad (86)$$

in which G^{\dagger} is only a regular inverse for the case of nine precision points and becomes a pseudoinverse for all other cases.

Analyzing the equation (86) for nine precision points and for an $\bar{\epsilon}$ with only small components, it is seen that the vector $\Delta \bar{h}$ will represent a vector which does satisfy all gradients of the matrix G and will cause the elements of the vector $\bar{\epsilon}$ to vanish. One could also give the vector $\Delta \bar{h}$ another meaning. If the components of all gradient vectors are taken in the direction of $\Delta \bar{h}$ and multiplied with the scalar $|\Delta \bar{h}|$, then they do form a vector equal to $-\bar{\epsilon}$. That is $-\bar{\epsilon}$ and those components of all dot products $\bar{g}_i \cdot \Delta \bar{h}$ as a vector must have the same direction if they were in the same vector space. This direction would be exactly equal to $-\bar{\epsilon}$ for nine or fewer precision points but will be only approximately equal if there are more than nine such points. The applied method yields a least square approximation and will be less accurate for a larger number of points. But it is held that the direction will always be in the direction of the negative $\bar{\epsilon}$ vector.

If, however, the n elements of the vector $\bar{\epsilon}$ are not small, then it is seen from equation (85) that the assumption $\epsilon_i^* = 0$ cannot be held any longer because, for larger elements in $\Delta\bar{h}$, one cannot ignore the higher order terms. Each equation of the n precision points should then be formulated as

$$\epsilon_i^* - \epsilon_i = -\Delta\epsilon_i = (\Delta\bar{h} \cdot \nabla)(\epsilon_i) + \text{higher order terms}$$

This shows that for those cases equation (86) would have to be changed to the form $\Delta\bar{\epsilon} = -G \cdot \Delta\bar{h}$ and would only be good for an unknown and for small $\Delta\bar{\epsilon}$. It indicates that for larger values of $\bar{\epsilon}$ the permissible values of $\Delta\bar{\epsilon}$ may also become too small to obtain an effective convergency and for those cases a different optimization method will be discussed in chapter VIII. But also for the well behaved case of small elements in $\bar{\epsilon}$ it was found that the vector $\Delta\bar{h}$ cannot be applied to its full extent and a search for its applicable step length had to be made. In general, it was found that $\Delta\bar{h}$ became most useful if its length remained rather small, which always was true for situations close to the desired absolute minimum.

Definition of Stepsize by the Use of a

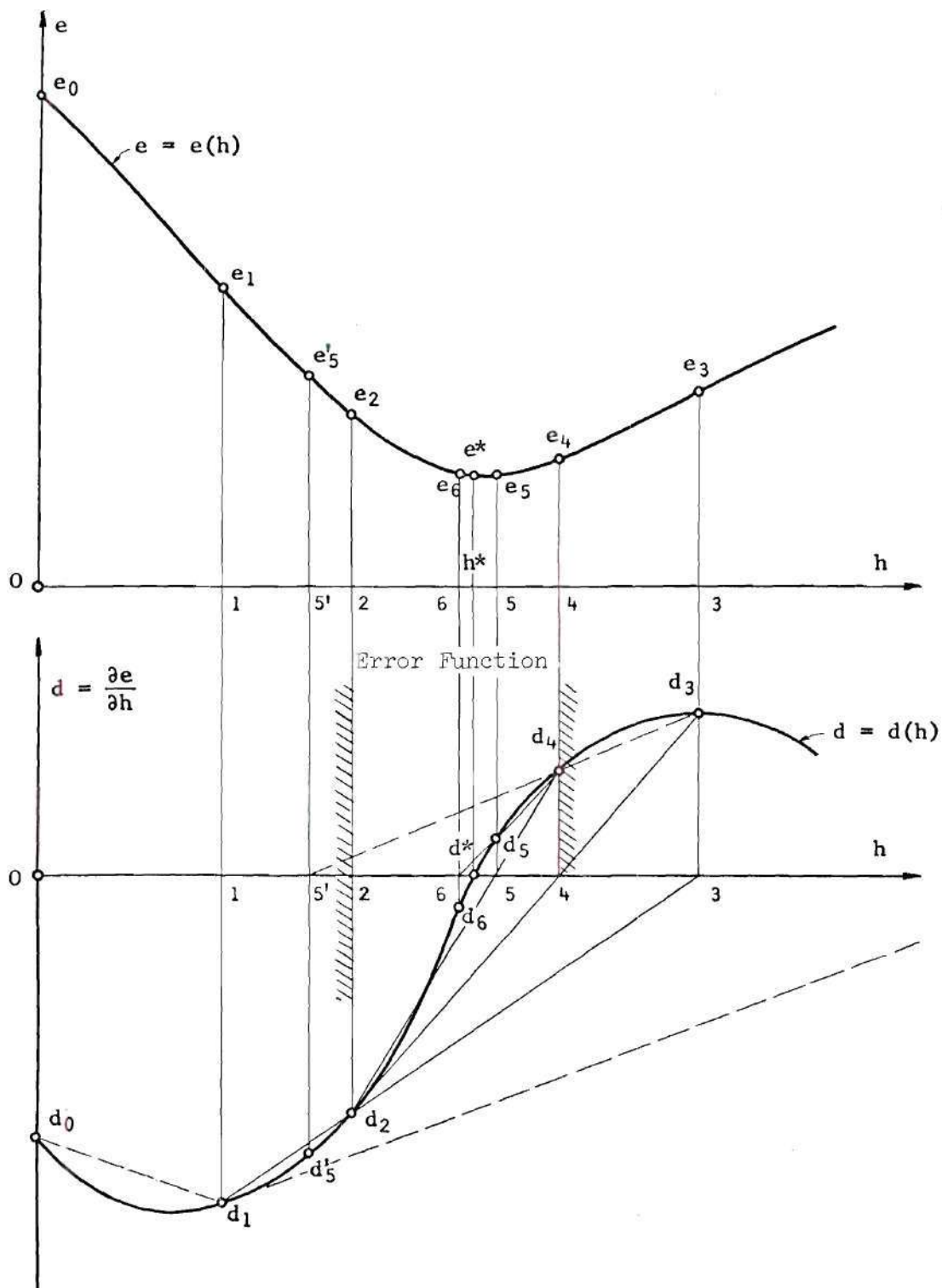
Modified Regula Falsi Method

Once the stepping vector $\Delta\bar{h}$ is found by the application of the gradient or the relaxation method, it is advantageous to make as much use of this vector as possible. It will point in the direction of lower values of the total error e and it is now the question to find out for what extent it can be applied.

Taking the direction of $\Delta\bar{h}$ as abscissa on which the length $h = |\Delta\bar{h}|$

can be plotted in relation to the total error e , it is possible to draw a diagram for the function $e = e(h)$ as shown in Figure 26. Here e is meant to be the sum of all errors ϵ_i^2 . The values ϵ_i were squared for making the iteration steps more effective for larger errors and to neglect the smaller ones. Hence, e will never be able to become negative but it could, for a perfect solution, become zero. In general, it will have a positive minimum which had to be found by the use of the differentiated curve, as shown underneath in Figure 26, and a modified regula falsi method.

The regula falsi method can only be effective for curves with a zero value as it is possible in the second diagram at the value $\frac{\partial e}{\partial h} = 0$ which becomes the condition for the minimum of the first diagram. Also the exceptional case in which the function for e becomes zero will have in this area a minimum and will still show a zero value in the derivative function of the second diagram. This permitted the use of the second curve for the detection of the relative minimum. At the same time, checks were made by the computer to stop the proceedings when no further improvement between the last iteration steps in the value for e was achieved, that is, when the absolute minimum was obtained. Here the difficulties in applying this method were of different nature than those shown in the paragraph on the Analog Simulation of chapter IV. Starting from the point 0 with a unit step along the abscissa h , a second point 1 is found for the calculation of the required next step leading towards the zero point. As one can see from Figure 26, the same precautions as in equation (26) of chapter IV were needed.



Derivative of Error Function

Figure 26. Modified "Regula Falsi" Method

$$\Delta h_{i+1} = |\Delta h_i| \cdot \frac{d_i}{|d_{i-1} - d_i|} \dots \text{ in which } d = \frac{\partial e}{\partial h} \quad (87)$$

The absolute signs were still needed because of the possibility of an increasing slope on the e-curve which could lead in the wrong direction, see points 0 and 1. And also the maximum step size had to be restricted.

No appropriate damping factor as in the previous application could be established and only a simple check was possible to reduce the oversize steps for Δh . Following the example of Figure 26, one can see that the point 2 would have gone out of range if no such check were made. The program will also check the signs of d at the points 1 and 2. If they are found to be of opposite sign, then the values for the variable h are set as the lower and upper boundaries between which the desired minimum for e is to be found. The next point 3 has such an opposite sign with the point 2 and will be memorized as the upper boundary and 2 as the lower one. Next the point 4 will be substituted for the point 3 as upper boundary because its d value is found also positive and smaller than the one of 3. At this moment, the boundaries will be at the points 2 and 4. The next computed point would be 5', which is outside the established boundaries. This will cause a swap of the point 3 against the previous point 2 and will result the point 5 at a much improved location. Because the new point 5 is still positive as 4 was, the upper boundary will be moved from 4 to 5.

In this fashion the boundaries together with the point position will be moved towards the desired minimum of the error sum e and the desired minimum will be reached faster than without the newly introduced check on the previous boundary position.

CHAPTER VIII

RELAXATION METHOD

General Comments

The main problem in the attempted solution remained the high non-linearity of the involved expressions which in these proceedings, to make the situation worse, cannot be expressed by analytical equations. Most texts on numerical analysis concentrate on linear relations expressed by matrix equations and sometimes include quadratic forms [58,59,73,74]; however, many quit when it comes to functions of higher order. The relaxation method seems to be originated by Gauss, who mentions it first in a letter to one of his students. He did not have any particular type of problem in mind but simply explained how easy it is to find the desired solution to a set of equations. It appears that he was also applying this method to linear equations only and that the application to nonlinear problems came much later. Dr. Stiefel, et al. [57] states (p. 105) that a comparison of many other iteration methods resulted in the discovery that the relaxation method seemed to work in all cases where other methods failed and it was also observed that in Russia more and more researchers rely on the relaxation method for general optimization problems [75].

It is peculiar for this method that it has no part of any high powered theoretical considerations and was simply found as a useful method which is simple to apply. As Gauss says, "It can be done while dreaming."

It has only one shortcoming which is to determine when the final solution is found. In simple applications, one can tell the end result by comparing the left side of an equation with the right side. If they match, the end result is reached. However, in the presented application the stipulation of the final end result is not that simple. The equations formulated here may or may not equate from one side to the other, and it becomes a delicate problem to determine when the best solution is reached. For this reason, it was decided to use the relaxation method for the initial approximation only and to use the gradient method whenever possible for the final condition.

Introduction to Relaxation

The principle of the relaxation method can be best demonstrated by the use of a set of linear equations of the form

$$A \bar{x} + \bar{a} = \bar{o}$$

This equation will not equate to \bar{o} (that is a zero vector) for an inadequate vector \bar{x} but will equate to a remainder vector \bar{r} .

$$A \bar{x} + \bar{a} = \bar{r} \tag{88}$$

The vector \bar{r} will vanish only for a perfect selection of \bar{x} and may never vanish for a nonsquare matrix A . The problem reduces now to the task of how to make the vector \bar{r} vanish and how to find a proper selection for \bar{x} . A could represent any type of matrix; however, using the relaxation method for a rectangular matrix with more rows than columns, it may become difficult to establish when the vector \bar{r} became a minimum.

The initial conditions of the mechanism allow here a better start than what is commonly in use, that is to start \bar{x} as a zero vector $\bar{0}$, which at the beginning will make \bar{r} equal to \bar{a} . The largest element of this initial vector $\bar{r}=\bar{a}$ can be found by inspection and also the largest element of that particular row in A can be spotted and a correction value Δx can be calculated by the simple relation

$$a_{ij}\Delta x_j = -r_i \quad \text{or} \quad \Delta x_j = -r_i/a_{ij} \quad (89)$$

The matrix element a_{ij} would be the largest row element of the i^{th} row which had the largest remainder element in \bar{r} . The remaining elements of \bar{x} are not altered, but the j^{th} element now is to be changed to $x_j + \Delta x_j$ and the result of this new vector \bar{x} is checked for its effect on \bar{r} in equation (88). It is obvious that in this case the introduced change in the i^{th} row will render a perfect solution for this row but not for the remaining rows of the matrix A. Next, there will be another row which will have the largest element in \bar{r} after the previous change of \bar{x} , and the process has to be repeated until a satisfactory solution for \bar{x} with an almost vanishing or perfectly vanishing \bar{r} is found.

This method may lead to a good solution with relatively few steps if the matrix A is well behaved but may lead to complications if A is not of that type, even in a simple linear case. Various texts show the required changes of the basic method which are called "Over-Relaxation" and "Group-Relaxation."

As there is no good analytical basis for this method, it should be noticed in the above description that there is no guarantee of any

type that when the i^{th} element of \bar{r} is corrected no other element will become just as large or even larger than the i^{th} element was before the change. And it is possible to build such simple problems in which this behavior will occur. It can well be that for a change in the i^{th} element the k^{th} one will become large and vice versa for the next change to reduce the k^{th} element again the i^{th} one will take its old size or become almost as large. It should be granted that the elements do have a tendency to become smaller and never become larger and this apparently is the only basis on which this system is successful. However, for such difficult cases, where the method does start to bounce back and forth as outlined, a group relaxation is suggested. This would mean to use two or three or more elements, whichever are larger compared to the others, instead of one element in the i^{th} row of A. These elements are then combined to a group for the correction of the i^{th} element in \bar{r} . Assuming three elements in the proposed group, this will change the equation (89) to

$$a_{i3} \Delta x_3 + a_{i5} \Delta x_5 + a_{i7} \Delta x_7 + \dots = -r_i$$

which should include all larger elements of A. But this equation has no unique solution for Δx and it is difficult to teach a computer to gain an experience element which will make "him" choose the best solutions for the group in \bar{x} .

Another idea is the overrelaxation which does purposely choose a too large correction element and in such a way does eliminate the critical area of the computations. Geometrically this would mean that the search for the solution became bounded by an area which excludes the de-

sired solution, and by a single overrelaxation was able to cross these boundaries and to continue the search outside in an area which was not accessible before.

In general, the experience shows that it is possible to produce a method in which the vector \bar{r} is gradually reduced to a minimum by directed changes of the vector \bar{x} . If such a method is available, it certainly becomes impractical to use a random selection of any type with no guidance at all for the next iteration step. Even a very poorly working relaxation method would represent a tremendous improvement as the result on page 106 shows. It appears that so far there were no outstanding efforts to provide the relaxation method with a well founded analytical and geometrical background which should prove the failure or success of its application and which should show what could be done for its improvement. It is not enough to show the fact that the method does work, an explanation should be offered to show why it is successful. It is unfortunate that mathematicians often disregard the physical meaning of a mathematical analysis and a geometric representation of a given problem.

For example, it may become possible to represent the relaxation method as a process which is feeling out the immediate neighboring area of a surface for valleys and follows these directions until the desired minimum is found. Such an end result will be effected by the nature of the surface, that is whether it is basically convex or concave. In the case of a concave surface, it could be said that no further reduction of the vector \bar{r} is possible once the absolute minimum is reached and any attempt for a further relaxation should result in an overall increase of the vector \bar{r} . It might well be that a resemblance to the differential

calculus could be established and would contribute for a better understanding of the present relaxation method.

Relaxation for Computer Use

When experimenting with path generators, it is soon realized that the given path will either be generated perfectly with all given precision points or it will be approximated with deviations forming groups of points. Each group of neighboring precision points will slightly miss the generated path either to one side or to the opposite side. This was actually to be expected after the experience with straight line generators. The pattern becomes especially apparent when approaching the final stages of the desired approximation. At the beginning of the operation, groups of distinct larger errors do exist and can easily be detected. As the synthesis of the path generator progresses with a successful relaxation, these protruding errors become smaller and are eventually almost of equal magnitude with the remaining ones. This, plus the fact that the relaxation method becomes more effective if applied to a group of larger errors instead of to a single point, leads to the difficulty of how to form these groups.

There may be other methods which will render good end results but the one used in here seemed to be very attractive in its simplicity. The integer "KT" used as a count for the successive iterations was included in the following logic expression

$$\text{IF } |\epsilon_{\max}| \cdot \frac{KT}{KT + 10} < |\epsilon_i| \text{ THEN } QC[j] \leftarrow i \quad (90)$$

First the maximum error ϵ_{\max} was found by a simple loop checking all

values for the errors ϵ_i . Then the above inequality is compared and if found true an account for the particular error number is placed in consecutive order into the integer array "QC". At the beginning, KT will be one and QC will show all index numbers of errors which were larger than $|\epsilon_{\max}| \cdot \frac{1}{11}$. As the convergency progresses, KT will become larger and, if for example equal to 20, QC will show only the numbers of errors which are larger than $|\epsilon_{\max}| \cdot \frac{2}{3}$. The attached quotient will approach the number one with increasing values for KT and the array QC will include only fewer error positions. This is exactly the feature desired in the relaxation process as the errors get more and more balanced in their magnitude. The speed by which this number approaches the value one can be controlled simply by changing the summation factor of the denominator.

After finding these largest errors, it was felt that a similar process should be applied to the coefficient matrix G, which is the same as in equation (86) used for the gradient method of the previous chapter. These elements of G are the components of the gradients for each precision point and indicate the effectiveness of one parameter change to a particular point. From this point of view, a large coefficient would mean a larger effect on the error and a small coefficient would mean almost no effect at all.

For this reason, it was planned to use the largest row elements of G only for the intended correction. But the results proved more efficient when all row elements were included of those rows with a larger error. This seemed to have a similar effect as the Group-Relaxation mentioned earlier. A simple least square solution was used for each

individual column with those row elements only which needed to be relaxed and which were found by equation (90). When returning to equation (89), the factors have to be changed from a_{ij} to g_{ij} , from Δx_j to Δh_j , and from r_i to ϵ_i . The factors g_{ij} are taken from the known gradient matrix G and the computed Δh_j will play the same role as it did before when calculated by using the gradients for each precision point. If i runs through all m indexes of the larger errors stored in the array QC , the least square solution becomes

$$\Delta h_j = -[\sum_{i=1}^m \epsilon_i g_{ij}] / \sum_{i=1}^m (g_{ij})^2$$

where j assumes all values from one to nine to include all nine parameters of the mechanism. The computed vector $\Delta \bar{h}$ is used for the required changes of the mechanism parameters after the value h^* is found by the modified regula falsi method as shown on page 114. The new parameter vector \bar{z} becomes

$$\bar{z}_r = \bar{z}_{r-1} + \Delta \bar{h} \cdot (h^*)$$

for the r^{th} iteration. Each iteration will require a new calculation of $\Delta \bar{h}$ and h^* .

Summarizing the above, the following points should be observed.

a. The matrix G is reduced to the rows which have the comparatively larger errors ϵ_i and in such a way a correction of the larger values is achieved.

b. The required corrections are computed independently for each mechanism parameter, as if it had to do the entire correction needed for

the chosen error group.

c. The selected least square method will also follow the relaxation idea which is a concentration on the larger values and a reduction of the smaller ones.

No special reasoning can be offered for the effectiveness of this method other than its correspondence to the known relaxation idea. That is to use a computed set of changes which should at least apply some corrections to the largest errors, hoping that the excluded smaller errors will not become too large at the same time. However, checking back once more into the gradient method, one can see from the Taylor series (see also equation (85) of chapter VII)

$$\bar{\epsilon}^* = \bar{\epsilon} + [G] \cdot \Delta \bar{h} + \text{higher order terms}$$

that for highly nonlinear functions it will almost never be possible to neglect the higher order terms because the assumption $\bar{\epsilon}^* = \bar{0}$ may require larger vectors for $\Delta \bar{h}$. In general, the steps $\Delta \bar{h}$ should always be kept small and the method will only become convergent near the desired optimum itself. The relaxation method, however, becomes effective for larger steps and for areas farther from the desired path optimum. It was further discovered that neither one of the methods above could solve a problem, but that it was best to apply an alternating system. That is, when one method reached an area in which the iteration did not make any further progress, to switch to the other one and in turn when this one did get stuck again to switch back to the first one. This idea should be in line with the so called overrelaxation and apparently does succeed in getting the iteration out of the deadlocked position even for the given highly nonlinear case.

CHAPTER IX

JOUKOWSKI PROFILE

General Comments

The complex number

$$z = x + iy \quad (91)$$

will represent the coordinates of a circle if the real numbers x and y are dependent on each other, for example by the relation $x^2 + y^2 = r^2$ which is a circle with its center in the coordinate origin, or by a relation of any other circle with its center in a more general location in the x - y plane. Equation (91) could also be considered as a transformation from real to complex coordinates. It shall be shown how it is possible to obtain an analytical and also a geometrical method for the construction of the Joukowski profile by making use of such a transformation.

Consider the complex variables z and ζ , and the real variables x , y , ξ , and η together with the transformation

$$\zeta = z + \frac{a^2}{z} \quad (92)$$

in which the complex variable $\zeta = \xi + i\eta$ will represent the Joukowski profile if the variable $z = x + iy$ does represent a circle which intersects the base circle k_b . The above transformation will allow one to relate the airfoil shape to a simple circular curve.

The graphical construction can best be explained by the representation of the complex number in terms of its norm $|z|$ and argument ϕ

such that

$$z = |z| \cdot e^{i\Phi} \quad \text{and} \quad 1/z = (1/|z|) \cdot e^{-i\Phi}$$

Introducing these forms into equation (92), it is possible to set

$$\zeta = z + (a^2/|z|) \cdot e^{-i\Phi}$$

which indicates that ζ becomes the sum of the vector z and a vector in a direction of the mirror image of z about the positive x-axis and of a length $a^2/|z|$.

Figure 27 shows two typical points P and Q for the construction of ζ with the points P' and Q' . The value a becomes the radius of the base circle k_b and the triangle OSP with a 90° angle at S permits the relation

$$|z'_P| = a^2/|z_P| \quad \text{for the point } P$$

and also the triangle OTN will have a relation of

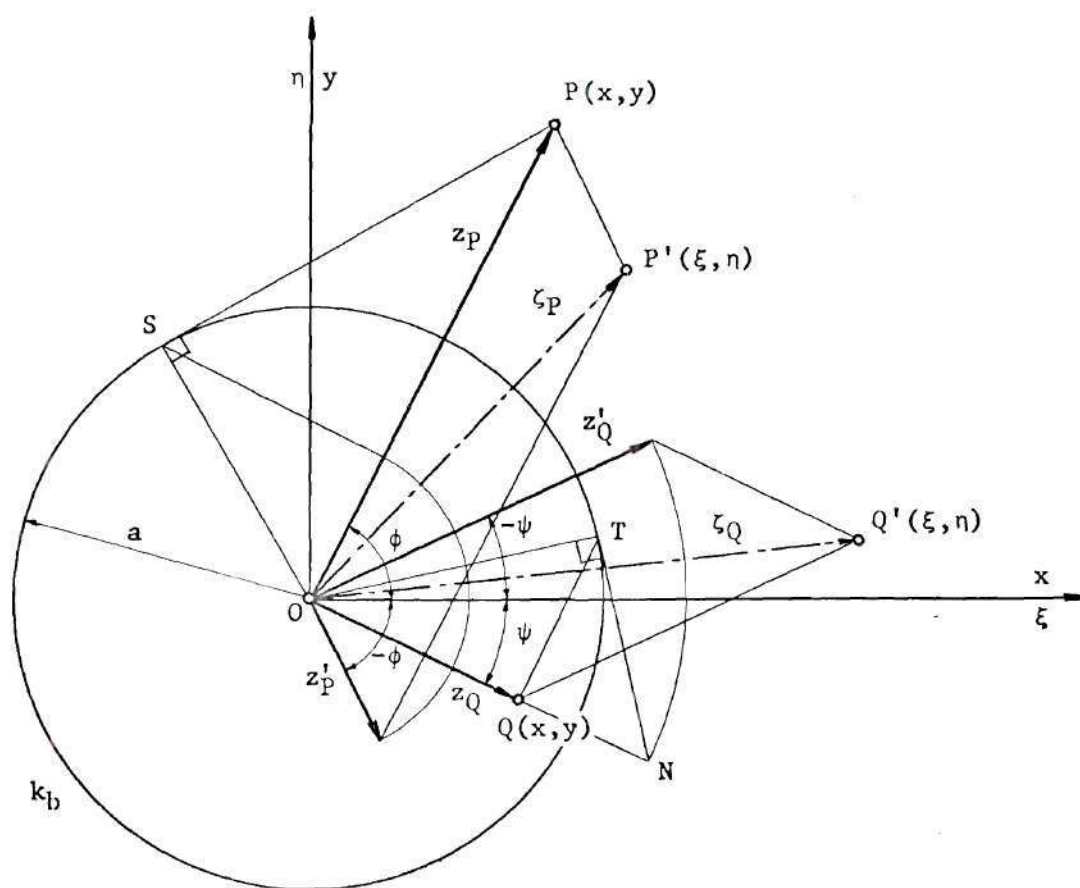
$$|z'_Q| = a^2/|z_Q| \quad \text{for the point } Q$$

Now with the lengths $|z'_P|$ and $|z'_Q|$ plotted in the mirror image directions the points for

$$z'_P = (a^2/|z_P|) \cdot e^{-i\Phi} \quad \text{and} \quad z'_Q = (a^2/|z_Q|) \cdot e^{-i\Psi}$$

are obtained. Finally, the two complex numbers z_P and z'_P are added as vectors to obtain ζ_P and the numbers z_Q and z'_Q to obtain the point Q' for ζ_Q .

This method would allow one to transfer graphically any group of points P into their new position P' for obtaining the new curve derived



k_b . . . base circle

P, Q . . points before the transformation

P', Q' . points after the transformation

x, y . . coordinates before the transformation

ξ, η . . coordinates after the transformation

$$\text{Transformation } \zeta = z + a^2/z$$

Figure 27. Graphical Representation of Transformation
 $\zeta = f(z)$

by the transformation in equation (92).

Analytically, it is possible to represent the transformed coordinates by the two relations

$$\xi = x \cdot \left(1 + \frac{a^2}{x^2 + y^2}\right) \quad \text{and} \quad \eta = y \cdot \left(1 - \frac{a^2}{x^2 + y^2}\right) \quad (93)$$

and there are three distinct cases worth mentioning which are defined by the positions of the points P in relation to the base circle k_b .

A. Base Line

Assume that the coordinates (x,y) of the point P coincide with the base circle and have the relation

$$x^2 + y^2 = a^2$$

then the coordinate transformation in equation (92) will give the values for

$$\xi = x \cdot \left(1 + \frac{a^2}{a^2}\right) = 2x \quad \text{and} \quad \eta = y \cdot \left(1 - \frac{a^2}{a^2}\right) = 0$$

which is a line on the x-axis with a length of 2a to either side of the coordinate origin. See the base line $\overline{R'P'}$ in Figure 28.

B. Core Line

The circle k_c with its center in M has the equation

$$x^2 + (y-s)^2 = r^2$$

and the radius r is related to the base circle radius a by

$$a^2 + s^2 = r^2$$

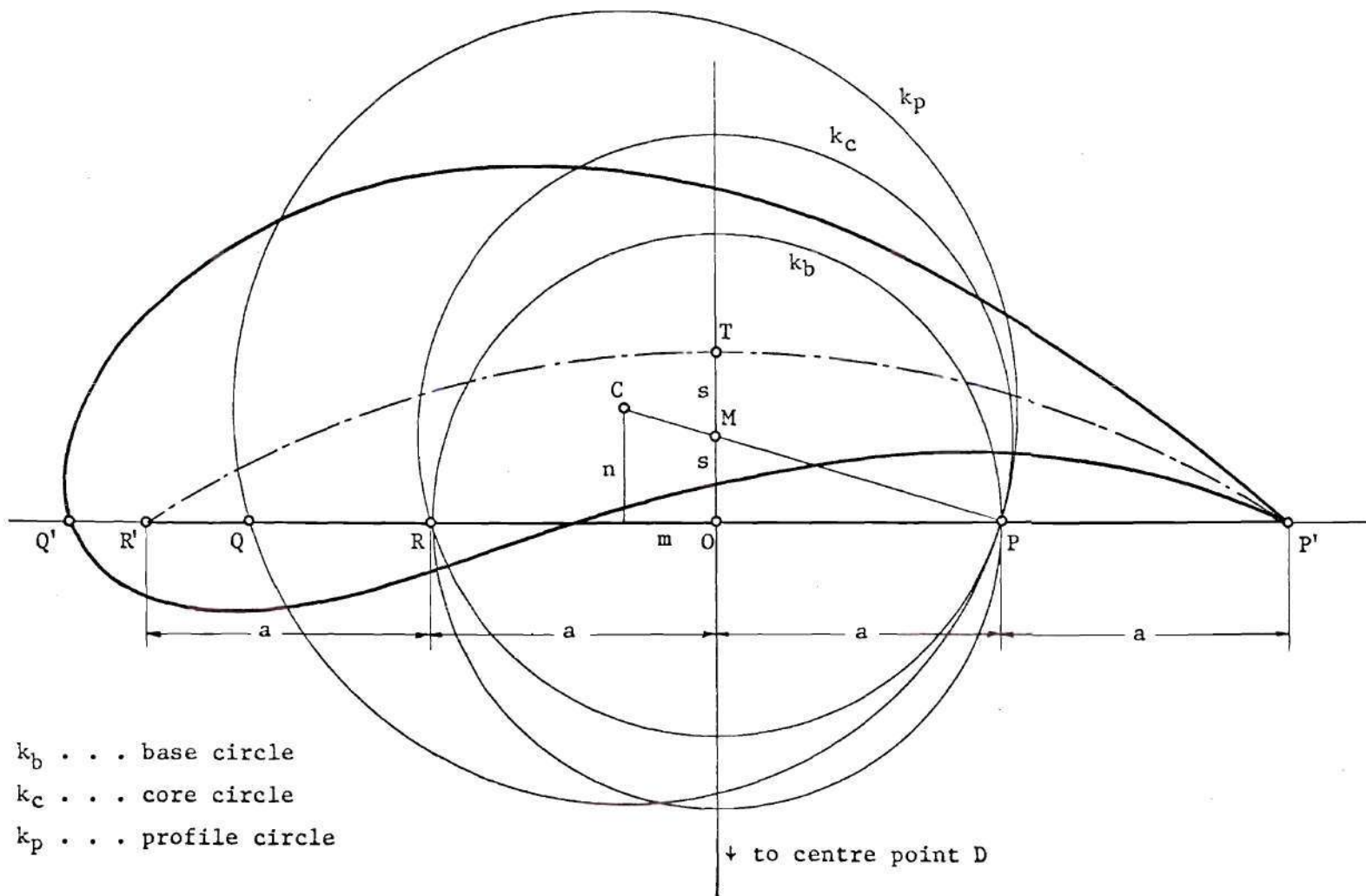


Figure 28. Joukowski Profile

which can be introduced into the above circle equation and will bring

$$x^2 + y^2 - 2sy = a^2 \quad (94)$$

First a few characteristic points of the core line. The intersection with the y-axis, the point T, will be found by setting $\xi=0$ in equation (93) and will show a solution for x in $x=0$ and η will become $\eta = y(1-a^2/y^2)$. But there is also from equation (94) $y^2 - 2sy = a^2$ because $x=0$ and will produce the relations $1 - a^2/y^2 = 2(s/y)$ and $\eta = y \cdot 2(s/y) = 2s$ which show that the intersection point T is at $\xi=0$ and $\eta=2s$.

The intersection with the x-axis P' and Q' can be found by setting $\eta=0$. This will bring two possibilities from equation (93), either $y=0$ or $a^2+y^2=a^2$. Equation (94) shows that both conditions must be filled at the same time and render the coordinates on the circle $x = \pm 0$ and $y=0$. The sought coordinate ξ becomes

$$\xi = \pm a \cdot (1 + a^2/a^2) = \pm 2a$$

Finally, it is claimed that the curve derived from the core circle is again a circle through the points T, P', and R'. If D is a circle center and q the distance \overline{OD} , then the square of the radius ρ of such a circle should equate to

$$\rho^2 = q^2 + 4a^2 = (q+2s)^2 \quad \text{or} \quad s^2 + sq = a^2 \quad (95)$$

From equations (93), (94), and (95), it is found that

$$\begin{aligned}
\xi^2 + (\eta + q)^2 &= \xi^2 + \eta^2 + 2\eta q + q^2 \\
&= x^2 \cdot \frac{(x^2 + y^2 + a^2)^2}{(x^2 + y^2)^2} + y^2 \cdot \frac{(x^2 + y^2 - a^2)^2}{(x^2 + y^2)^2} + 2qy \cdot \frac{x^2 + y^2 - a^2}{x^2 + y^2} + q^2 \\
&= \frac{(x^2 + y^2 - a^2)^2}{x^2 + y^2} + \frac{4a^2 x^2}{x^2 + y^2} + 2qy \cdot \frac{x^2 + y^2 - a^2}{x^2 + y^2} + q^2 \\
&= 4y^2 \cdot \frac{s^2 + qs}{x^2 + y^2} + 4a^2 \cdot \frac{x^2}{x^2 + y^2} + q^2 = 4a^2 + q^2 = (q + 2s)^2 \\
&= \rho^2 = \text{const.}
\end{aligned}$$

which proves the circular shape of the core line.

C. Airfoil Profile

If the circle k_p is chosen with its center in the point C of the coordinates $(-m, n)$, the equation of the circle becomes

$$(x+m)^2 + (y-n)^2 = x^2 + y^2 + 2xm - 2yn + m^2 + n^2 = r^2$$

and the Joukowski profile has then the coordinates

$$\xi = \frac{x}{x^2 + y^2} \cdot (x^2 + y^2 + a^2) \qquad \eta = \frac{y}{x^2 + y^2} \cdot (x^2 + y^2 - a^2)$$

Because of the relation $r^2 = n^2 + (m+a)^2$, the circle equation can be changed to

$$x^2 + y^2 + 2xm - 2yn = a(2m+a)$$

Again, it is of interest to find the points P' and Q' for $\eta=0$. This will make $y=0$ or $x^2 + y^2 = a^2$. The above equation for the circle k_p shows that x cannot become $-a$ but has the two values

$$x_1 = a \quad \text{and} \quad x_2 = -(2m+a)$$

Substituting these results into equation (93) will give

$$\xi_1 = x_1 + a^2/x_1 = 2a$$

$$\xi_2 = \frac{x_2^2 + a^2}{x_2} = \frac{-2 \cdot (2m^2 + 2ma + a^2)}{2m+a}$$

$$= -2 \cdot (2m+a - 2m \cdot \frac{m+a}{2m+a}) < -2 \cdot (2m+a - 2m) = -2a$$

but the same value can also be approximated by

$$\xi_2 = x_2 + a^2/x_2 = -(2m+a + \frac{a^2}{2m+a}) > -(2m+a+a) = -2(m+a)$$

which indicates an estimated location for ξ_2 between

$$-2a > \xi_2 > -2(m+a)$$

A similar process would show that the point T is not in the center of the profile along the y-axis. But still, the curve through R', T and P' does take a core position in the profile because, if the circle k_p is made smaller by moving its center C towards M but keeping its periphery point P fixed, then the profile will become thinner until it eventually will coincide with the core line itself.

It should be observed that the point P must be a periphery point of the circles k_b and k_p if the profile should have a cusp at P'. But it cannot be made to have two cusps, one at each end, because then it will become a single line only and will not show a thickness any more.

Actual Profile Coordinates

With the known circle transformation, the method for calculating any number of coordinates for the Joukowski profile can be best demonstrated by introducing the angle α for the coordinates of the input circle k_p .

$$x = r \cos(\alpha) - m \quad \text{and} \quad y = r \sin(\alpha) + n$$

The coordinates of the profile become

$$\xi = x \left(1 + \frac{a^2}{x^2 + y^2}\right) \quad \text{and} \quad \eta = y \left(1 - \frac{a^2}{x^2 + y^2}\right)$$

The radius r can be calculated from the relation

$$r = \sqrt{n^2 + (m+a)^2}$$

The computer calculated these values in a small procedure which used the relation $n = (m+a) \cdot s/a$ and was arranged with a DO statement with C the number of points desired and PO an array for all profile coordinates.

```
PROCEDURE JOUK(A,S,M,C,PO)
```

```
  N ← (M+A) × S/A
```

```
  Δα ← 2π/(C-1) ; α1 ← -ARCTAN(S/A)
```

```
  PO[1,1] ← PO[1,C] ← 2 × A ; PO[2,1] ← PO[2,C] ← 0
```

```
  R ← SQRT(N2 + (M+A)2)
```

```
  FOR i←2 STEP 1 UNTIL C-1 DO ↓
```

```
    αi ← αi-1 + Δα
```

```
    X ← R × COS(αi) - M ; Y ← R × SIN(αi) + N
```

```
    DI ← A2/(X2 + Y2)
```

```
    PO[1,i] ← X × (1+DI) ; PO[2,i] ← Y × (1-DI)
```

The first point in the array PO will be the same as the last one

in order to ensure a closed curve for the path generated by the four-bar mechanism.

Profile Synthesis

The effectiveness of the combined relaxation and gradient method is demonstrated by the synthesis of a Joukowski profile. In order to make efficient use of the available computer time, only nine precision points were selected for the preliminary iteration steps. They were measured from the sketch shown in Figure 29 and are listed in Table 2. The points were distributed to obtain a good accuracy at important profile sections. The initial coupler curve c_0 was chosen from a first guess but had to have approximately the same extensions as the desired profile. This was needed to eliminate any confusing results from the error definition. However, it was found that the cusp end of the profile still presented a difficult section because it will easily permit two different error readings. The progressing approximation, as shown in curve c_{30} , permits an error reading to the lower curve section but also one to the upper section. Both coupler point positions can be obtained accidentally because they follow in the mechanism motion so close to each other. The previous approximation shown as c_{18} has a similar alternate error reading possibility, but they cannot be confused because they are much farther apart in the angular motion of the driver link and therefore are easier to control. If such alternate readings occur, it is possible for the differentiation procedure to become a continuous loop and will require special attention.

Figure 30 shows the initial mechanism in dotted lines and Table 1

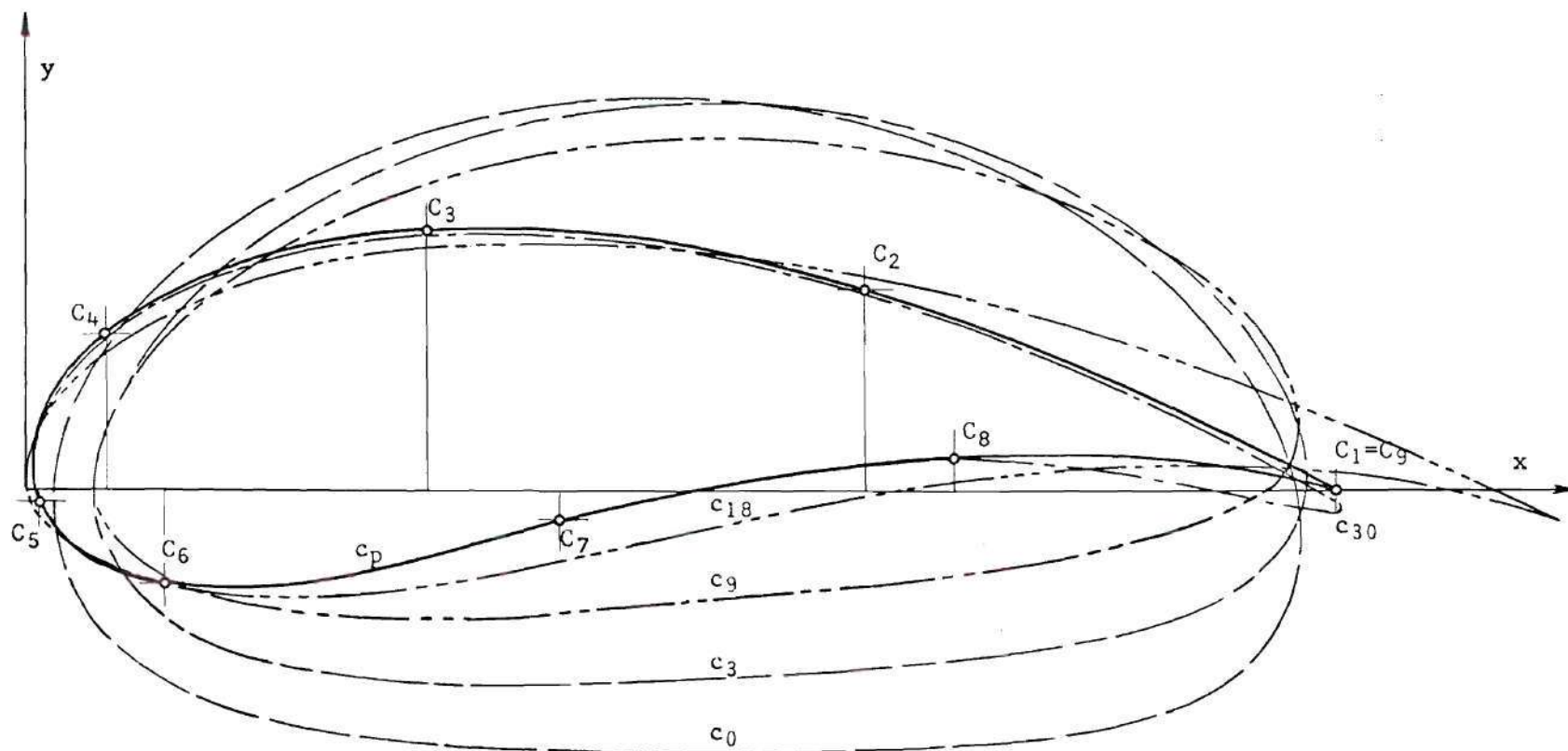


Figure 29. Preliminary Synthesis of Joukowski Profile

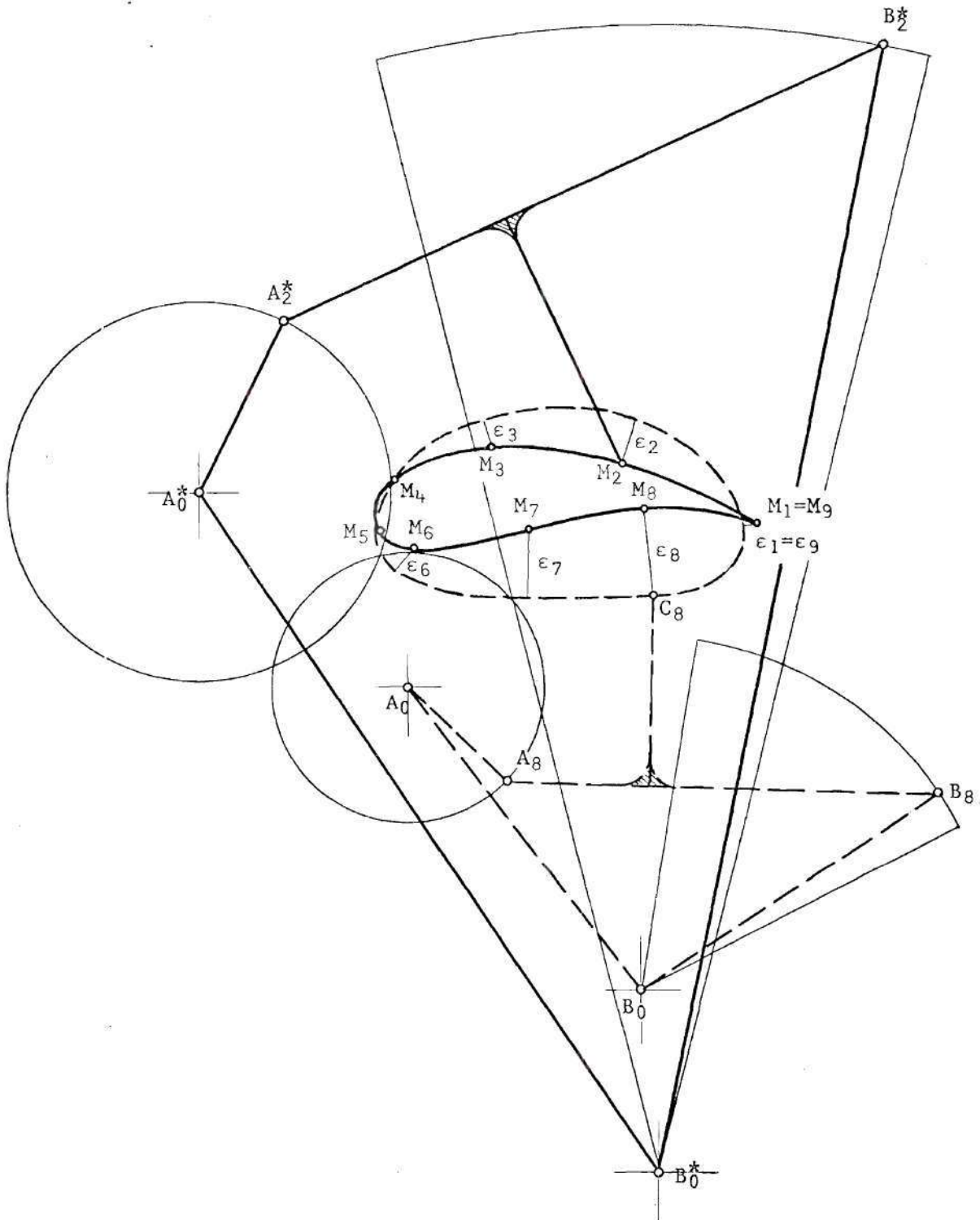


Figure 30. Airfoil Path Generation, Initial and Final Stages

Table 1. Changes of Link Dimensions for Nine Precision Points

Iteration	Link Dimensions								
	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	$\lambda = z_9$
0	12.50	4.45	13.91	11.60	4.71	-6.18	8.65	-15.30	126.87
3	13.56	4.35	14.88	11.83	6.47	-6.26	10.31	-14.97	137.09
9	15.50	3.94	19.81	20.01	10.69	-7.30	9.78	-20.74	133.94
18	29.37	7.21	22.72	34.47	12.41	10.22	10.80	-19.78	131.20
30	28.82	6.12	15.27	34.64	7.06	7.00	9.51	-23.80	116.76

Table 2. Coordinates and Approximation Error of the Precision Points

Coordinates of Precision Points									
Point No.	1	2	3	4	5	6	7	8	9
x	12.42	7.98	3.81	0.76	0.15	1.30	5.10	8.83	12.42
y	0.0	1.89	2.42	1.47	-0.10	-0.85	-0.30	0.30	0.0
Iteration	Individual Precision Point Error								
0	-0.373	1.496	0.914	-0.064	-0.068	0.790	2.141	2.679	-0.373
3	-0.382	1.634	0.766	-0.404	-0.596	0.126	1.379	1.741	-0.382
9	-0.467	1.317	0.430	-0.457	-0.538	-0.031	0.865	1.122	-0.467
18	-0.218	0.161	-0.138	-0.164	0.016	-0.004	0.273	0.182	-0.218
30	-0.039	-0.004	-0.010	-0.010	-0.002	0.003	0.002	-0.001	-0.039

Table 3. Link Dimensions for
Thirty Precision
Points

Link No.	Link Dimensions	
	Begin	Itera- tion 7
z ₁	27.170	26.750
z ₂	6.167	6.198
z ₃	17.618	21.368
z ₄	34.812	37.607
z ₅	7.547	7.858
z ₆	7.798	8.908
z ₇	9.281	9.238
z ₈	-21.918	-21.235
z ₉	120.424	123.636

Table 4. Coordinates and Errors for
Thirty Precision Points

No.	Coordinates		Error	
	X	Y	Begin	Itera- tion 7
1	12.42	0.0	-0.010	-0.001
2	12.23	0.08	0.019	0.026
3	11.73	0.32	0.046	0.051
4	11.0	0.69	0.047	0.053
5	10.09	1.11	0.026	0.033
6	9.06	1.54	-0.008	0.002
7	7.94	1.93	-0.041	-0.029
8	6.78	2.22	-0.064	-0.051
9	5.62	2.41	-0.072	-0.058
10	4.48	2.46	-0.062	-0.049
11	3.41	2.38	-0.037	-0.026
12	2.44	2.18	-0.003	0.005
13	1.59	1.87	0.028	0.034
14	0.91	1.47	0.046	0.052
15	0.40	1.02	0.037	0.045
16	0.10	0.55	-0.008	0.003
17	0.01	0.10	-0.068	-0.056
18	0.15	-0.28	-0.090	-0.084
19	0.51	-0.57	-0.047	-0.052
20	1.10	-0.73	0.033	0.016
21	1.93	-0.74	0.099	0.073
22	2.98	-0.61	0.111	0.084
23	4.26	-0.36	0.055	0.036
24	5.73	-0.05	-0.035	-0.042
25	7.34	0.23	-0.084	-0.089
26	8.96	0.38	-0.046	-0.060
27	10.41	0.37	0.036	0.013
28	11.53	0.23	0.091	0.066
29	12.21	0.07	0.042	0.069
30	12.42	0.0	-0.010	0.048
Abs. Totals:			0.298	0.273

shows in the row, headed by iteration zero, its dimensions by using the same symbols as shown in Figure 24. Table 2 lists the coordinates of the nine precision points and the initial errors. Figure 29 also shows a few sample curves, illustrating the approximation after the third, the ninth, the eighteenth, and the thirtieth iterations, and they are also listed in Tables 1 and 2. It can be seen that the approximation progressed very well but was not able to sense the lower part of the profile end as shown by the curve c_{30} . This shortcoming was finally remedied by the addition of further precision points which were not included initially to save computer time.

After receiving this approximate synthesis of the airfoil profile, the "Jouk" procedure, described earlier, was used to calculate 30 point coordinates along a Joukowski profile with the values for

$$A \leftarrow 3 ; \quad S \leftarrow 0.575 ; \quad M \leftarrow 0.65$$

which were roughly of the same profile as the one used for the initial nine point approximation. The initial and final link dimensions are shown in Table 3, and a sketch of the final mechanism is also represented in Figure 30. Table 4 shows the individual coordinates of all thirty points and the error readings at the beginning and after seven iterations. The overall totals of all errors indicate the seven iterations were enough to improve the approximation by roughly 10 percent. Figure 31 shows the very good fit of the synthesized curve against the actual profile and shows seven intersecting points between the generated and the desired curve. There is one additional point which cannot be found from the changing error signs in Table 4 but can be seen on the sketch

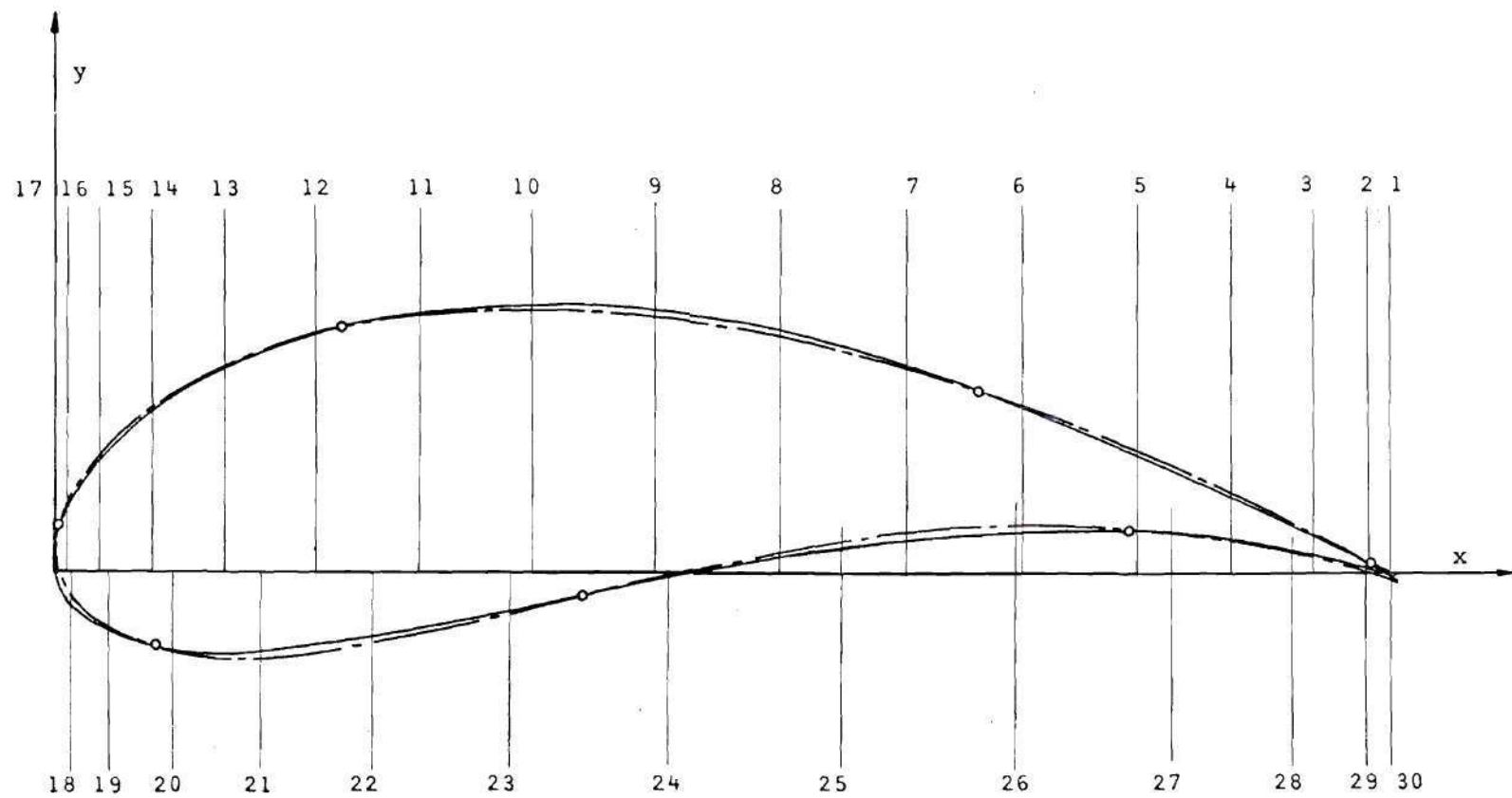


Figure 31. Thirty Precision Point Synthesis of Joukowski Profile

at the profile end. It is known that any two closed curves must intersect at an even number of points. But it was also established that, if the Joukowski profile does not represent a perfect coupler curve, it can have not more than nine intersections. Therefore, the seven marked intersections with the eighth unmarked one at the profile end indicate already a good approximation for the desired profile.

It should be understood that with the increased number of point coordinates the computer time has to be extended by a multiple and it becomes a question of time allowance to bring the approximation to a point where no further reduction of the error can be obtained or the calculated gradient vector becomes a zero vector. This result was obtained with a simpler nine precision point problem in which the iteration process was stopped by the computer itself after a coupler curve was obtained which went exactly through the selected points and where no further improvement was possible.

CHAPTER X

COMPUTER PROGRAM

General Routines

The computer program was written in Extended Algol for the Burroughs' B-5500 and consisted basically of various procedures which were called by a rather short main program but also by other procedures. The grouping of the procedures was partly caused by the repeated use but also by the development process of the entire program in various sections.

First, a special square root routine was needed to eliminate small negative arguments which stopped the computer calculations but merely represented a zero value with an accumulated negative random error.

REAL PROCEDURE SRT(X)

SRT ← IF (X<0 AND -X<@-7) OR X<@-9 THEN 0 ELSE SQRT(X)

This will set the square root equal to zero for small negative arguments but should also help to reduce some of the positive random errors which are expected to be of a different magnitude.

Similar reasons lead to another procedure representing the difference between the cosine function and one.

REAL PROCEDURE COM(X)

COM ← IF ABS(X)<.2 OR ABS(π -X)<.2 THEN -2×SIN²(X/2) ELSE
COS(X)-1

There was also a need for a function which will produce the angle for any point on the unit circle in all four quadrants. The available arcustangent function of the computer can only produce positive or negative results which are smaller than $\pi/2$ taken as absolute value.

REAL PROCEDURE ARG(X,Y)

AX ← ABS(X)>1.5@-9 ; AY ← ABS(Y)>1.5@-9

IF AX AND AY THEN

AG ← ARCTAN(Y/X)

ARG ← IF X<0 THEN AG+ π ELSE
IF Y<0 THEN AG+2 π ELSE AG

ELSE

IF AX THEN

ARG ← IF X>0 THEN 0 ELSE π

ELSE

IF AY THEN

ARG ← IF Y>0 THEN $\pi/2$ ELSE $3\pi/2$

ELSE

ARG ← 0

The mobility or motion limits are determined by another procedure which can work out either the motion limits of the driver or of the follower by simply interchanging the positions for the values Z_2 and Z_4 and by using the negative supplementary angle of ψ instead of ϕ .

PROCEDURE SWIKL($Z_1, Z_2, Z_3, Z_4, \phi, BO, \alpha_1, \alpha_2$)

$\phi' \leftarrow$ IF $\phi > \pi$ THEN $\phi - 2\pi$ ELSE

($-\pi < \phi' < \pi$)

IF $-\phi \geq \pi$ THEN $\phi + 2\pi$ ELSE ϕ

BO1 ← $Z_1 + Z_2 \leq Z_3 + Z_4$; BO2 ← $ABS(Z_1 - Z_2) \geq ABS(Z_3 - Z_4)$

IF BO1 AND BO2 THEN GO TO L2

IF BO1 THEN GO TO L1

```

X ← ((Z3+Z4)2 - Z12 - Z22) / (2×Z1×Z2) ; Y ← SIGN(φ')×SQRT(1-X2)
δ ← ARG(X,Y)
IF δ>π THEN δ ← δ-2π (-π<δ<π)
α1 ← δ-φ'
IF BO2 THEN
  α2 ← -δ-φ'+2π×SIGN(φ')
  GO TO L2
L1: X ← ((Z3-Z4)2 - Z12 - Z22) / (2×Z1×Z2) ; Y ← SIGN(φ')×SQRT(1-X2)
δ ← ARG(X,Y)
IF δ>π THEN δ ← δ-2π (-π<δ<π)
α2 ← δ-φ'
IF BO1 THEN
  α1 ← -δ-φ'
L2: BO ← NOT (BO1 AND BO2)
IF ABS(α1)<1.5@-9 THEN α1 ← 0
IF ABS(α2)<1.5@-9 THEN α2 ← 0

```

The Boolean BO will indicate whether or not there were any limitations and the angles α_1 and α_2 will show the amount of those limitations measured from the link position indicated by the angle ϕ .

Four Accuracy Point Method

This procedure called ACPONT could be used in place of the READ statement for the initial mechanism dimensions and it follows, with the exception of the end portion, the same computations as shown in Sandor's publication [38]. If it is used, a few subroutines are to be included and are listed below. The use of complex variables which are available with some computer languages would simplify many of the quoted statements.

The first subroutines of this list is the procedure SWINKL. It establishes the angular motion range and is almost the same as the procedure SWIKL, however, it has a few modifications to allow the use of a general coordinate system which is not in line with the frame link. The array Z contains the coordinates of the mechanism links. The Booleans B01 and B02 will indicate the detected mobility of the mechanism and the array OTP will contain its restricting boundaries.

PROCEDURE SWINKL(ZN,B01,B02,OTP)

```

ZR1 ← SQRT(Z1 + ZN[1,1]2 + ZN[2,1]2)
ZR2 ← SQRT(Z2 + ZN[1,2]2 + ZN[2,2]2)
ZR3 ← SQRT(Z3 + ZN[1,3]2 + ZN[2,3]2)
ZR4 ← SQRT(Z4 + ZN[1,4]2 + ZN[2,4]2)
β ← ARG(ZN[1,2], ZN[2,2]) - ARG(ZN[1,1], ZN[2,1])
IF β ≤ (-π) THEN β ← β + 2π ELSE IF β > π THEN β ← β - 2π      (-π < β < π)
B01 ← (ZR1 + ZR2) ≤ (ZR3 + ZR4)
B02 ← ABS(ZR1 - ZR2) ≥ ABS(ZR3 - ZR4)
IF B01 AND B02 THEN GO TO L2
IF B01 AND NOT B02 THEN GO TO L1
X ← ((ZR3 + ZR4)2 - Z1 - Z2) / (2 × ZR1 × ZR2) ; Y ← SIGN(β) × SQRT(1 - X2)
δ ← ARG(X, Y)
IF δ > π THEN δ ← δ - 2π      (-π < δ < π)

```

```

      OTP[1] ←  $\delta - \beta$  ; OTP[3] ←  $-\delta - \beta$ 
      OTP[5] ←  $2\pi \times \text{SIGN}(\beta) + \text{OTP}[3]$ 
      IF NOT BO1 AND BO2 THEN GO TO L2
    →L1: X ←  $((\text{ZR3} - \text{ZR4})^2 - \text{Z1} - \text{Z2}) / (2 \times \text{ZR1} \times \text{ZR2})$  ; Y ←  $\text{SIGN}(\beta) \times \text{SQRT}(1 - X^2)$ 
       $\delta \leftarrow \text{ARG}(X, Y)$ 
      IF  $\delta > \pi$  THEN  $\delta \leftarrow \delta - 2\pi$  ( $-\pi < \delta < \pi$ )
      OTP[2] ←  $\delta - \beta$  ; OTP[4] ←  $-\delta - \beta$ 
    →L2: END

```

The procedure ANGL will determine all possible output angles including those of the symmetry position for a given input angle of the compatibility "linkage". The input angle is recorded in the array INP[1]. The array DT contains the needed values for the minors of the augmented matrix which form the compatibility "linkage". The two remaining positions of the array INP will be filled with two additional angles which are derived from the motion the coupler and follower of the compatibility "linkage" have due to the displacement caused by the first input angle. The array OTP will receive the angles for the same linkage system but for the symmetry position and will carry in its first position the same angle as in the first position of INP. Here, it was not possible to use a coordinate system parallel to the frame link and, therefore, a few extra statements are needed to account for all possible configurations.

PROCEDURE ANGL(INP,DT,OTP)

```

      CW ← COS(INP[1]) ; SW ← SIN(INP[1])
      DR ← DT[1,1] + DT[1,2] × CW - DT[2,2] × SW
      DI ← DT[2,1] + DT[2,2] × CW + DT[1,2] × SW
      DT3 ← DT[1,3]2 + DT[2,3]2 ; DT4 ← DT[1,4]2 + DT[2,4]2
      D ← DR2 + DI2
      X ← (DT4 - DT3 - D) / (2 × SQRT(DT3 × D))

```



```

 $\alpha_3 \leftarrow \text{ARG}(X, \text{SQRT}(1-X^2))$ 
 $X \leftarrow (DT3-DT4-D) / (2 \times \text{SQRT}(DT4 \times D))$ 
 $\alpha_4 \leftarrow \text{ARG}(X, \text{SQRT}(1-X^2))$ 
 $AR3 \leftarrow \text{ARG}(DT[1,3], DT[2,3])$ 
 $AR4 \leftarrow \text{ARG}(DT[1,4], DT[2,4])$ 
 $ARD \leftarrow \text{ARG}(DR, DI)$ 
 $OTP[1] \leftarrow INP[1]$ 
IF  $AR3 < ARD$  THEN  $AR3 \leftarrow AR3 + 2\pi$ 
 $INP[2] \leftarrow ARD - AR3 + \alpha_3$  ;  $OTP[2] \leftarrow ARD - AR3 - \alpha_3 + 2\pi$ 
IF  $AR4 > ARD$  THEN  $ARD \leftarrow ARD + 2\pi$ 
 $INP[3] \leftarrow ARD - AR4 - \alpha_4$  ;  $OTP[3] \leftarrow ARD - AR4 + \alpha_4 - 2\pi$ 

```

The procedure WINKEL is used for the needed selection of compatible positions for the compatibility "linkage" and will include several solutions as they may become possible in various angular ranges. The array INP has to contain the selected angular displacements. The array FU is filled by the values $\cos(\Delta\phi)-1$ and $\sin(\Delta\phi)$ computed from the displacement angles listed in INP. The array OTP becomes a list of the desired first displacement angle of the coupler or the follower with all alternate solutions and the integer variable KN gives the number of solutions shown in OTP. The variable RAT permits the selection of the angle in its permissible range. The array DT is filled with all minor determinants of the augmented matrix (See Fig. 33). And finally the array DL is needed which must contain the coordinates of the coupler point displacement vectors.

PROCEDURE WINKEL(INP, FU, OTP, RAT, DT, DL, KN)

FOR $i \leftarrow 1$ STEP 1 UNTIL 3 DO

$FU[1,i] \leftarrow \text{COM}(INP[i])$ $FU[2,i] \leftarrow \text{SIN}(INP[i])$
--

$DT[1,1] \leftarrow DT[2,1] \leftarrow 0$

the position for the first accuracy point. The array PSA holds the computed angles for ψ . The arrays FU and FV contain the required cosine and sine functions for the angles ϕ and θ as they were calculated in the procedure WINKEL. DS and DT are arrays with the minor determinants of the augmented matrices and are also obtained from the procedure WINKEL. The array RP holds the coordinates of the four selected precision points and DL represents again the coordinates of the displacement vectors. K is an integer which gives an account for the number of solutions.

PROCEDURE SOL(PSA,Z,FU,FV,DS,DT,RP,DL,K)

CS1 ← COM(PSA[1]) ; SS1 ← SIN(PSA[1])

CS2 ← COM(PSA[2]) ; SS2 ← SIN(PSA[2])

DLR5 ← FV[1,1]×FU[1,2] - FV[2,1]×FU[2,2] - FV[1,2]×FU[1,1]
+ FV[2,2]×FU[2,1]

DLI5 ← FV[1,1]×FU[2,2] + FV[2,1]×FU[1,2] - FV[1,2]×FU[2,1]
- FV[2,2]×FU[1,1]

DLR6 ← FV[1,1]×CS2 - FV[2,1]×SS2 - FV[1,2]×CS1 + FV[2,2]×SS1

DLI6 ← FV[1,1]×SS2 + FV[2,1]×CS2 - FV[1,2]×SS1 - FV[2,2]×CS1

DL5 ← DLR5²+DLI5² ; DL6 ← DLR6²+DLI6²

IF DL5<@-9 OR DL6<@-9 THEN GO TO L1

DLR3 ← DL[1,1]×CS2 - DL[2,1]×SS2 - DL[1,2]×CS1 + DL[2,2]×SS1

DLI3 ← DL[2,1]×CS2 + DL[1,1]×SS2 - DL[2,2]×CS1 - DL[1,2]×SS1

K ← K+1

Z[1,2,K] ← (DT[1,4]×DLR5 + DT[2,4]×DLI5)/DL5

Z[2,2,K] ← (DT[2,4]×DLR5 - DT[1,4]×DLI5)/DL5

Z[1,4,K] ← (DT[1,4]×DLR6 + DT[2,4]×DLI6)/DL6

Z[2,4,K] ← (DT[2,4]×DLR6 - DT[1,4]×DLI6)/DL6

Z[1,5,K] ← (-DS[1,4]×DLR5 - DS[2,4]×DLI5)/DL5

Z[2,5,K] ← (-DS[2,4]×DLR5 + DS[1,4]×DLI5)/DL5

Z[1,6,K] ← (DLR3×DLR6 + DLI3×DLI6)/DL6

Z[2,6,K] ← (DLI3×DLR6 - DLR3×DLI6)/DL6

```

FOR i+1 STEP 1 UNTIL 2 DO ↓
  Z[i,3,K] + Z[i,5,K] - Z[i,6,K]
  Z[i,1,K] + Z[i,4,K] - Z[i,2,K] - Z[i,3,K]
  Z[i,7,K] + RP[i,1] - Z[i,6,K] - Z[i,4,K]
FOR i+8 STEP 1 UNTIL 10 DO ↓
  Z[1,i,K] + PSA[i-7]
→L1: END

```

The procedure LINK will calculate all available solutions of the mechanism for one computed set of the angles θ . The array THA has to hold the values for θ . The coordinates for the linkage system with its position for the first accuracy point will be received in the three dimensional array Z. RAT is a proportionality factor which is required for the procedure WINKEL. The array DS has to hold the values for the minor determinants used in the procedure SOL. FU is the same array as it was in the procedure WINKEL. RP holds all precision point coordinates and DL all displacement vector coordinates. K is an integer accounting for the number of solutions found by the procedure SOL.

PROCEDURE LINK(THA,Z,RAT,DS,FU,RP,DL,K)

WINKEL(THA,FV,PSA,RAT,DT,DL,KN)

FOR i+1 STEP 1 UNTIL 2*KN DO ↓

```

KR + K
SOL(PSA[i,*],Z,FU,FV,DS,DT,RP,DL,K)
IF KR ≠ K THEN ↓
  FOR j+8 STEP 1 UNTIL 10 DO Z[2,j,K] + THA[j-7]

```

The last procedure of this group ACPONT is the actual four accuracy point routine and is used to find a solution for the selected input angles. It will select from a given list of desired precision points a

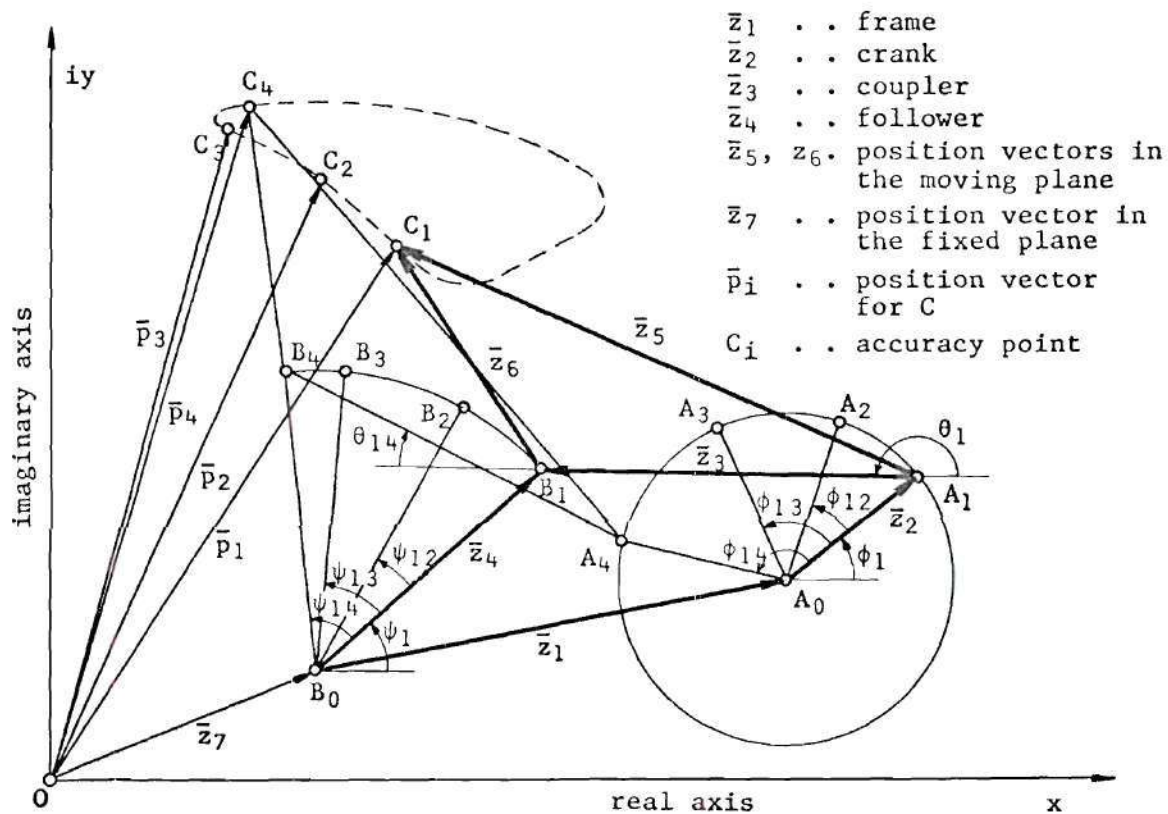


Figure 32. Vectors in Complex Number Notation

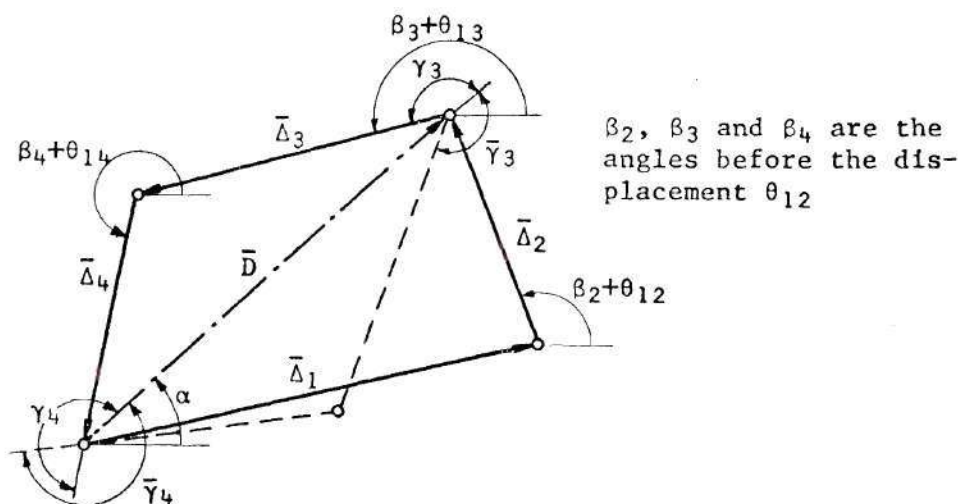


Figure 33. Compatibility Configuration

set of four points. It will compute a few possible solutions and select of those the ones with all four chosen points on one closed coupler curve. However, it was not possible, at this point, to include a check for the obtained point sequence. But the solutions are also screened for the proportions of the link dimensions. Mechanisms with a large ratio between the longest and the shortest link are discarded and the one with the most balanced proportions is finally selected.

The array R contains the coordinates of all precision points. PHA is an array for all chosen displacement angles of the driver. RAT is a ratio to permit a choice for two additional angles in the computations. The array ZA will receive the desired mechanism dimensions and the variables ϕ and ψ will position this mechanism for the first accuracy point; they are the angles for the driver and the follower in relation to the frame. The integer N indicates the number of precision points contained in R.

PROCEDURE ACPONT(R,PHA,RAT,ZA, ϕ , ψ ,N)

K ← N DIV 4

FOR i←1 STEP 1 UNTIL 2 DO FOR j←0 STEP 1 UNTIL 3 DO

RP[i,j+1] ← R[i,j×K+1]

FOR i←1 STEP 1 UNTIL 2 DO FOR j←2 STEP 1 UNTIL 4 DO

DL[i,j-1] ← RP[i,j] - RP[i,1]

→L1: WINKEL(PHA,FU,THA,RAT,DS,DL,K)

M ← 0

FOR i←1 STEP 1 UNTIL 2×K DO

LINK(THA[i,*],Z, RAT,DS,FU,RP,DL,M)

(M is the number of solutions shown in the array Z with the coordinates of the mechanism links as shown in Fig. 32)

KN ← 0

FOR i+1 STEP 1 UNTIL M DO (begin of the solution evaluation)

FOR j+1 STEP 1 UNTIL 2 DO

FOR k+1 STEP 1 UNTIL 4 DO

$ZN[j,k] \leftarrow Z[j,k,i]$

SWINKL(ZN,BO1,BO2,LIM)

$N \leftarrow 0$

IF NOT BO1 AND NOT BO2 THEN

$GA1 \leftarrow LIM[1] ; GA2 \leftarrow LIM[2]$

ELSE

IF NOT BO1 AND BO2 THEN

$GA1 \leftarrow LIM[1] ; GA2 \leftarrow LIM[5]$

ELSE

IF BO1 AND NOT BO2 THEN

$GA1 \leftarrow LIM[4] ; GA2 \leftarrow LIM[2]$

ELSE

GO TO L2

IF $GA2 > 0$ THEN

FOR j+1 STEP 1 UNTIL 3 DO

IF $PHA[j] > GA2$ OR $PHA[j] < GA1$ THEN $N \leftarrow 1$

ELSE

FOR j+1 STEP 1 UNTIL 3 DO

IF $PHA[j] < GA2$ OR $PHA[j] > GA1$ THEN $N \leftarrow 1$

GO TO IF $N=0$ THEN L3 ELSE L4

→L2:

FOR j+1 STEP 1 UNTIL 2 DO

$ZN[j,1] \leftarrow Z[j,1,i]$

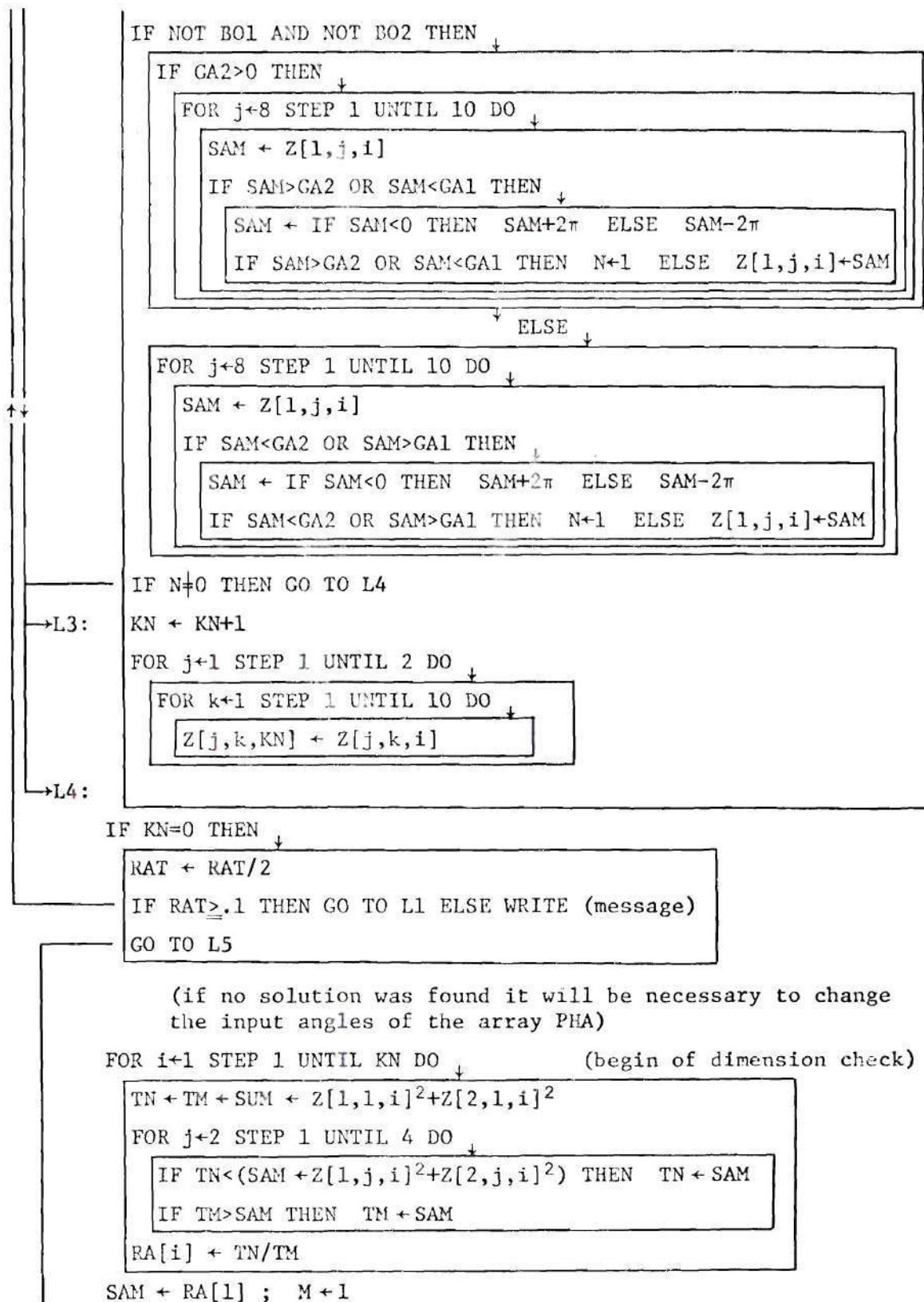
$ZN[j,2] \leftarrow Z[j,4,i]$

$ZN[j,3] \leftarrow -Z[j,3,i]$

$ZN[j,4] \leftarrow -Z[j,2,i]$

SWINKL(ZN,BO1,BO2,LIM)

$GA1 \leftarrow LIM[1] ; GA2 \leftarrow LIM[2]$



FOR i←2 STEP 1 UNTIL KN DO

IF SAM>(SUM+RA[i]) THEN

SAM ← SUM ; M ← i

(the choosen mechanism is transferred into the array ZA, see Fig. 12)

FOR i←1 STEP 1 UNTIL 5 DO

ZA[i] ← SQRT(Z[1,i,M]²+Z[2,i,M]²)

ZA[9] ← IF (RAT ← ARG(Z[1,1,M],Z[2,1,M]))>π THEN RAT-2π ELSE RAT
(-π<RAT<π)

β ← ARG(Z[1,3,M],Z[2,3,M]) - ARG(Z[1,5,M],Z[2,5,M])

ZA[6] ← ZA[5]×SIN(β) ; ZA[5] ← ZA[5]×COS(β)

ZA[7] ← Z[1,7,M] ; ZA[8] ← Z[2,7,M]

φ ← IF (φ ← ARG(Z[1,2,M],Z[2,2,M]) - ZA[9])>π THEN φ-2π ELSE φ

ψ ← ARG(Z[1,4,M],Z[2,4,M]) - ZA[9] (-π<φ<π)

→L5: END

General Matrix Inversion

The matrix inversion program is a matrix manipulation which permits the determination of the matrix rank and of the independent rows or columns. The input matrices are A and B, where B becomes a unit matrix if A is to be inverted. The result is obtained in the matrix ANS. However, if B is a simple vector then ANS becomes also a vector which satisfies the equation $A \times \overline{ANS} = \overline{B}$. The integer NR is the rank of the Matrix A and the elements of the integer array DEP indicate the index of all independent rows or columns. The integer N represents the required basic array size of A and the integer Q accounts for the number of columns in B. The array space of A must permit one additional row for tracking any interchanges of the columns.

PROCEDURE INV(A,N,B,Q,ANS,DEP,NR)

FOR j+1 STEP 1 UNTIL N DO

A[N+1,j] ← j

FOR i+1 STEP 1 UNTIL N DO

FOR j+1 STEP 1 UNTIL Q DO
ANS[i,j] ← B[i,j]

FOR k+1 STEP 1 UNTIL N DO

(start of algorithm)

IF k=N THEN GO TO L1

TEST ← ABS(A[k,k])

R ← k

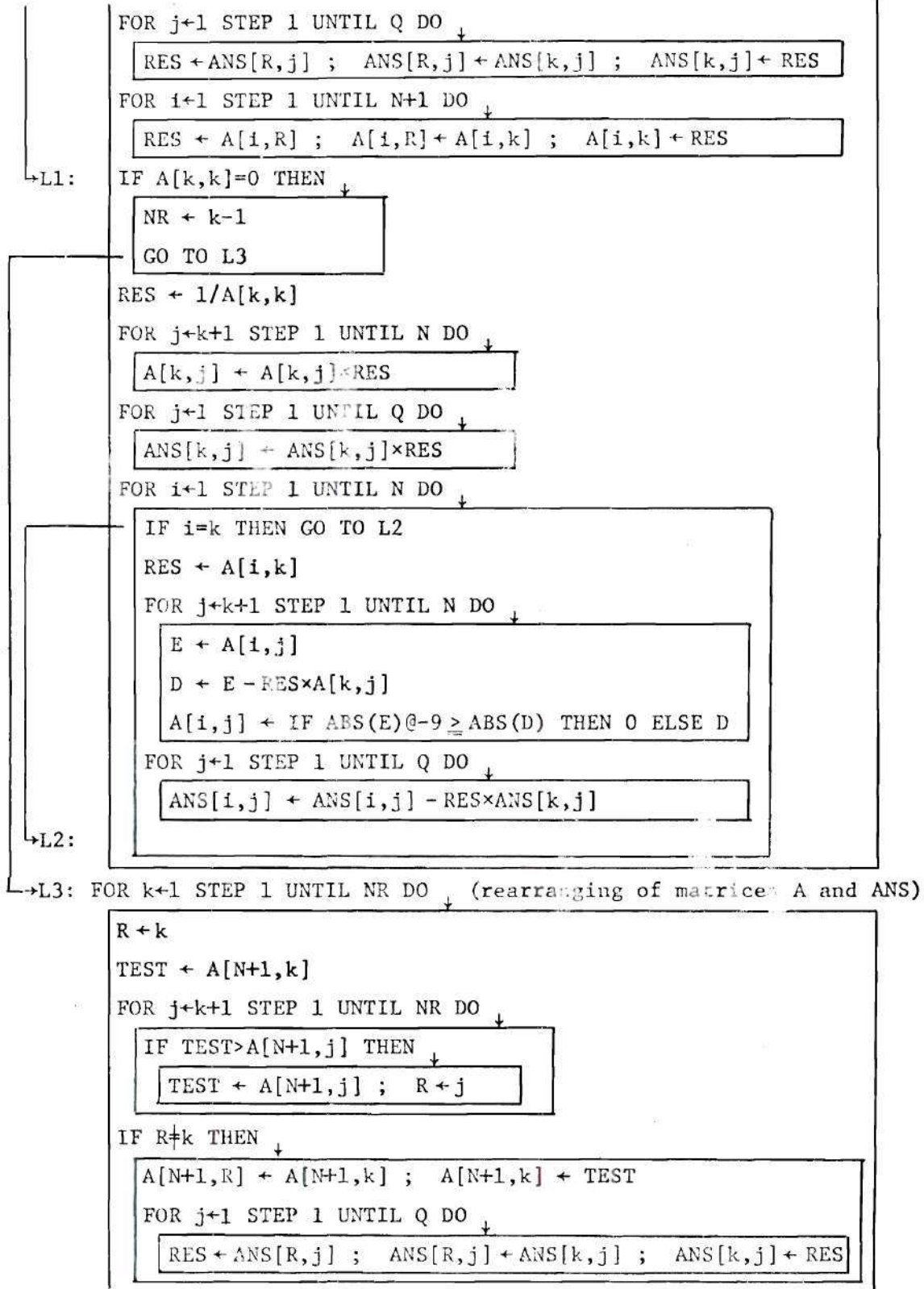
FOR j←k+1 STEP 1 UNTIL N DO

IF TEST < ABS(A[j,j]) THEN
TEST ← ABS(A[j,j]) ; R ← j

IF R=k THEN GO TO L1

FOR j←k STEP 1 UNTIL N DO

RES ← A[R,j] ; A[R,j] ← A[k,j] ; A[k,j] ← RES



```

FOR j+1 STEP 1 UNTIL NR DO
  DEP[j] ← A[N+1,j]
IF N=Q AND NR<Q THEN
  FOR j+1 STEP 1 UNTIL NR DO
    K ← DEP[j]
    FOR i+1 STEP 1 UNTIL NR DO
      ANS[i,j] ← ANS[i,K]

```

The least square method for the inversion of rectangular matrices requires an additional procedure in which AR represents the matrix, M the number of rows, N the number of columns and PR and XR are vectors to satisfy the relation $AR \times \overline{XR} = \overline{PR}$. For a given \overline{PR} it is possible to approximate the relation by an \overline{XR} which is computed by the least square method. The shown method may not be the simplest one but becomes very versatile when a few extras are added. The approximation of the vector \overline{PR} can be changed to receive a few elements exact and to approximate the remaining ones by the least square. In order to help the operation of the computer the array AR was inverted to carry the larger number M for the row elements in its second position and the smaller number N in its first position but the procedure will also work for cases where M becomes equal to N.

PROCEDURE LSQ(AR,M,N,PR,XR)

```

FOR k+1 STEP 1 UNTIL N DO
  FOR j+1 STEP 1 UNTIL N DO
    SUM ← 0
    FOR i+1 STEP 1 UNTIL M DO
      SUM ← SUM + AR[k,i] × AR[j,i]
    A[k,j] ← SUM

```

(represents $[A^T A]$)

FOR i+1 STEP 1 UNTIL M DO ↓

FOR j+1 STEP 1 UNTIL M DO ↓

Q[i,j] ← IF i=j THEN 1 ELSE 0

(represents [I])

INV(A,N,Q,N,RES,DRUP,NO)

(RES = $[A^T A]^{-1}$)

IF NO=N THEN ↓

(routine for nonsingular matrix)

IF M=N THEN ↓

(inversion of square matrix)

FOR i+1 STEP 1 UNTIL M DO ↓

B[i,1] ← PR[i]

FOR j+1 STEP 1 UNTIL N DO ↓

A[i,j] ← AR[j,i]

INV(A,N,B,1,RES,DRUP,NO)

FOR j+1 STEP 1 UNTIL N DO ↓

XR[j] ← RES[j,1]

GO TO L2

FOR k+1 STEP 1 UNTIL N DO ↓

FOR i+1 STEP 1 UNTIL M DO ↓

SUM ← 0

FOR j+1 STEP 1 UNTIL N DO ↓

SUM ← SUM + RES[k,j] × AR[j,i]

AT[k,i] ← SUM

($AT = A^+ = [A^T A]^{-1} A^T$)

GO TO L1

FOR k+1 STEP 1 UNTIL NO DO ↓

J ← DRUP[k]

FOR i+1 STEP 1 UNTIL M DO ↓

B[i,k] ← AR[J,i]

FOR k+1 STEP 1 UNTIL NO DO ↓

FOR i+1 STEP 1 UNTIL M DO ↓

SUM ← 0

FOR j+1 STEP 1 UNTIL NO DO ↓

SUM ← SUM + RES[k,j] × B[i,j]

BT[k,i] ← SUM

($BT = B^+ = [B^T B]^{-1} B^T$)

FOR k←1 STEP 1 UNTIL M DO

FOR i←1 STEP 1 UNTIL M DO

SUM ← 0

FOR j←1 STEP 1 UNTIL N DO

SUM ← SUM + AR[j,k] × AR[j,i]

A[k,i] ← SUM

$$(A = [AA^T])$$

INV(A,M,Q,M,RES,DROP,NO)

FOR k←1 STEP 1 UNTIL NO DO

I ← DROP[k]

FOR j←1 STEP 1 UNTIL N DO

C[k,j] ← AR[j,I]

FOR k←1 STEP 1 UNTIL NO DO

FOR j←1 STEP 1 UNTIL N DO

SUM ← 0

FOR i←1 STEP 1 UNTIL NO DO

SUM ← SUM + C[i,j] × RES[i,k]

CT[j,k] ← SUM

$$(CT = C^+ = C^T [CC^T]^{-1})$$

FOR k←1 STEP 1 UNTIL NO DO

S ← DRUP[k]

FOR i←1 STEP 1 UNTIL N DO

SUM ← 0

FOR j←1 STEP 1 UNTIL NO DO

SUM ← SUM + CT[i,j] × C[j,S]

AI[i,k] ← SUM

$$(AI = C^+ H^{-1})$$

FOR k←1 STEP 1 UNTIL N DO

FOR j←1 STEP 1 UNTIL M DO

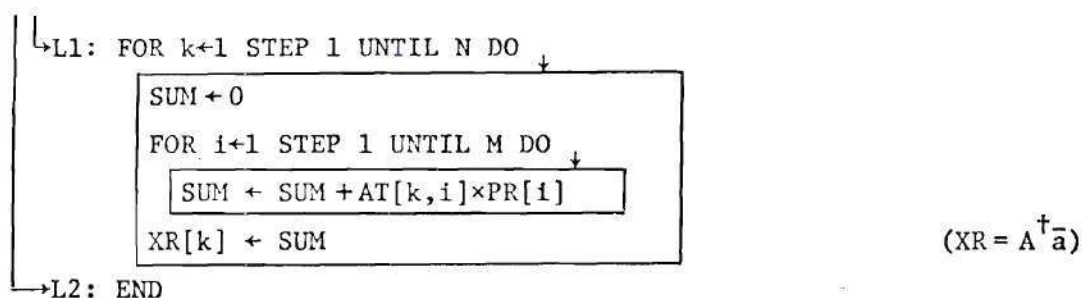
SUM ← 0

FOR i←1 STEP 1 UNTIL NO DO

SUM ← SUM + AI[k,i] × BT[i,j]

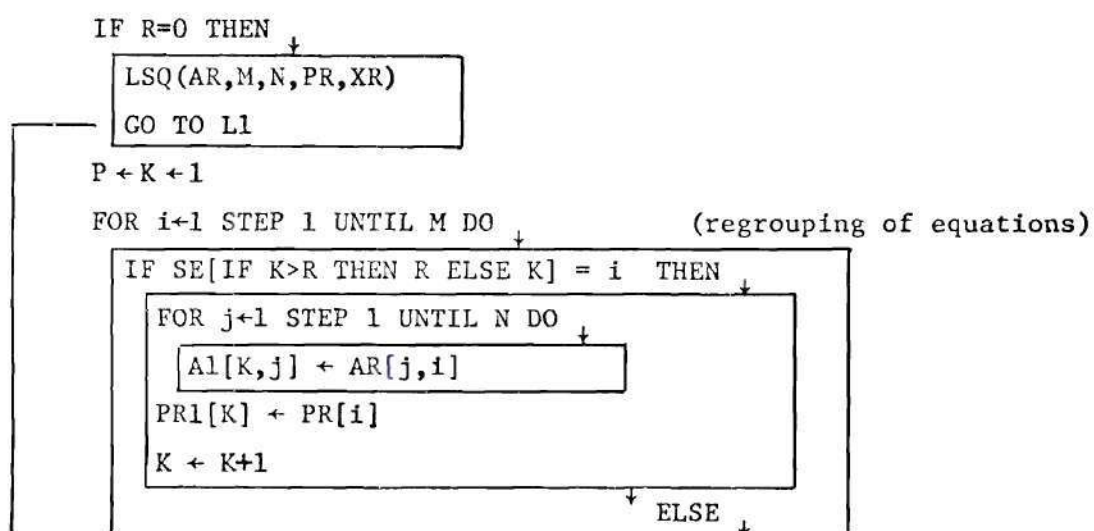
AT[k,j] ← SUM

$$(AT = A^+ = C^+ H^{-1} B^+)$$



Finally, to accomplish the selective approximation, an additional procedure is introduced in which AR represents the basic matrix with its dimensions M and N as before in the procedure LSQ. The array PR is a vector which represents the inhomogeneous part of the equations of which those listed in the integer array SE are to be supplied with an exact solution. The number R must indicate the number of equations listed in SE. The output is obtained in XR which represents a single row vector. This last routine takes advantage of the inversion routine and can adapt itself to any type of column dependency of the coefficient matrix AR as long as there is a mathematical solution possible.

PROCEDURE LSQF(AR,M,N,SE,R,PR,XR)



FOR j+1 STEP 1 UNTIL N DO

A3[P,j] ← AR[j,i]

PR2[P] ← PR[i]

P ← P+1

MR ← M-R ; NR ← N-R

FOR i+1 STEP 1 UNTIL N DO

FOR j+1 STEP 1 UNTIL N DO

T[i,j] ← IF i=j THEN 1 ELSE 0

(represents [I])

FOR j+1 STEP 1 UNTIL N DO

FOR i+1 STEP 1 UNTIL N DO

SUM ← 0

FOR k+1 STEP 1 UNTIL R DO

SUM ← SUM + A1[k,i] × A1[k,j]

A[i,j] ← SUM

($A = [A_1^T A_1]$)

INV(A,N,T,N,RES,DROP,NO)

(testing the rank)

IF NO < R THEN

WRITE(message)

GO TO L1

P ← K + 1

FOR i+1 STEP 1 UNTIL N DO

(grouping of equations)

IF DROP[IF K > R THEN R ELSE K] = i THEN

FOR j+1 STEP 1 UNTIL R DO

A1[j,k] ← A1[j,i]

FOR j+1 STEP 1 UNTIL MR DO

A3[j,k] ← A3[j,i]

K ← K+1

ELSE

FOR j+1 STEP 1 UNTIL R DO

A2[j,P] ← A1[j,i]

FOR j+1 STEP 1 UNTIL MR DO

A4[j,P] ← A3[j,i]

P ← P+1

INV(A1,R,T,R,RES,DRUP,NO)

(RES = [A1]⁻¹)

FOR i+1 STEP 1 UNTIL MR DO

FOR j+1 STEP 1 UNTIL NR DO

SAM ← 0

FOR k+1 STEP 1 UNTIL R DO

SUM ← 0

FOR s+1 STEP 1 UNTIL R DO

SUM ← SUM + RES[k,s] × A2[s,j]

SAM ← SAM + A3[i,k] × SUM

E ← A4[i,j]

G ← E - SAM

Q[j,i] ← IF ABS(E) × @-9 ≥ ABS(G) THEN 0 ELSE G

(A₄ - A₃A₁⁻¹A₂)

FOR i+1 STEP 1 UNTIL MR DO

SAM ← 0

FOR j+1 STEP 1 UNTIL R DO

SUM ← 0

FOR k+1 STEP 1 UNTIL R DO

SUM ← SUM + RES[j,k] × PR1[k]

SAM ← SAM + A3[i,j] × SUM

E ← PR2[i]

G ← E - SAM

QR[i] ← IF ABS(E) × @-9 ≥ ABS(G) THEN 0 ELSE G

(ā₂ - A₃A₁⁻¹ā₁)

LSQ(Q,MR,NR,QR,XR2)

FOR i+1 STEP 1 UNTIL R DO

SAM ← 0

FOR j+1 STEP 1 UNTIL R DO

SUM ← 0

FOR k+1 STEP 1 UNTIL NR DO

SUM ← SUM + A2[j,k] × XS2[k]

E ← PR1[j]

G ← E - SUM

G ← IF ABS(E) × @-9 ≥ ABS(G) THEN 0 ELSE G

SAM ← SAM + RES[i,j] × G

XR1[i] ← SAM

(A₁⁻¹[ā₁ - A₂ā₂])

P ← K ← 1

FOR i ← 1 STEP 1 UNTIL N DO

IF DROP[IF K > R THEN R ELSE K] = 1 THEN

XR[i] ← XR1[K]

K ← K + 1

ELSE

XR[i] ← XR2[P]

P ← P + 1

→ L1: END

Analog Computer Simulation

The procedure which actually produces the error readings is the one which follows the coupler point motion and can be considered as an analog computer simulation. The required input values are XP and YP the coordinates of the desired precision point, an array ZA containing all physical dimensions of the mechanism including its position and angular orientation, α_1 and α_2 the two limitations of the rotation of the driver, a Boolean BO1 characterizing the motion limitations and finally i the index number of the precision point. The output from the procedure is ϵ the measured error between the desired and the obtained coupler point position in a direction normal to the obtained coupler curve and further the angles ϕ_i and ψ_i indicating the end positions of the driver and the follower links. See Fig. 20.

PROCEDURE ERROR(XP,YP,ZA, α_1 , α_2 ,BO1,i, ϵ , ϕ_i , ψ_i)

OWN REAL δ_1, δ_2

(rotation of coordinates)

$XM \leftarrow XP \times \cos(Z_9) + YP \times \sin(Z_9)$; $YM \leftarrow -YP \times \sin(Z_9) + YP \times \cos(Z_9)$

$K \leftarrow 1$; $p \leftarrow \dot{\theta} \leftarrow 0$

(checking of the mechanism position with a possible correction and setting of time variable; ϕ , θ and ψ are global variables)

$TM \leftarrow \text{TIME}(2)$

IF i=1 THEN

IF BO1 THEN

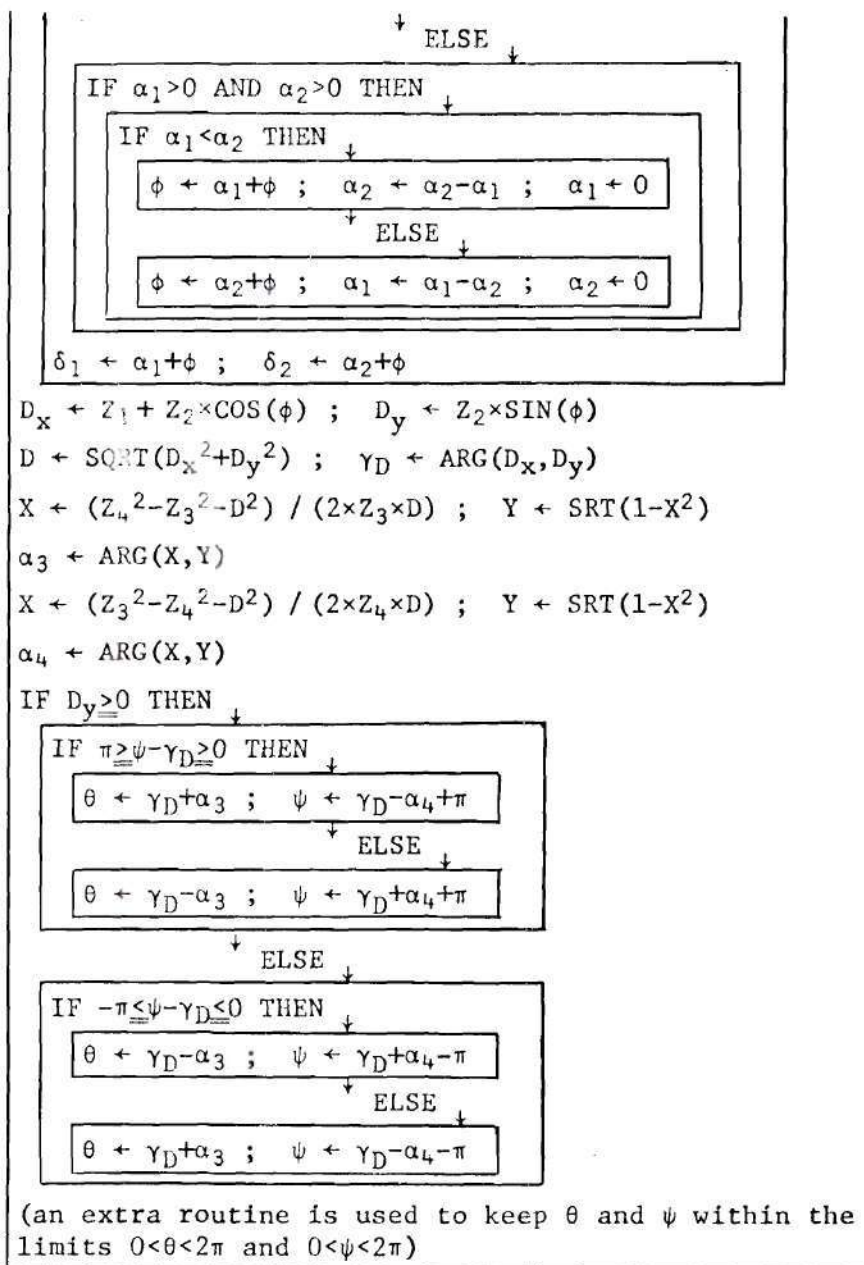
IF $\alpha_1 < 0$ AND $\alpha_2 < 0$ THEN

IF $\alpha_1 < \alpha_2$ THEN

$\phi \leftarrow \alpha_2 + \phi$; $\alpha_1 \leftarrow \alpha_1 - \alpha_2$; $\alpha_2 \leftarrow 0$

ELSE

$\phi \leftarrow \alpha_1 + \phi$; $\alpha_2 \leftarrow \alpha_2 - \alpha_1$; $\alpha_1 \leftarrow 0$



DIF $\leftarrow \theta - \psi$

SIG $\leftarrow \text{SIGN}(\text{DIF})$

→L1: (this section is repeated until the desired coupler point position is obtained or the time constant is exceeded)

$p^* \leftarrow p ; \dot{\theta}^* \leftarrow \dot{\theta}$

$X_T \leftarrow Z_5 \times \cos(\theta) + Z_6 \times \sin(\theta) ; Y_T \leftarrow Z_5 \times \sin(\theta) - Z_6 \times \cos(\theta)$

$\phi' \leftarrow \text{IF } \phi < 0 \text{ THEN } \phi + 2\pi \text{ ELSE } \phi \quad (0 < \phi' < 2\pi)$

IF ABS(SIN(DIF)) < @-8 THEN

IF ABS(DIF) < @-7 THEN

$\dot{X} \leftarrow -YT \times \text{SIGN}(\Delta\psi) \times \text{SIGN}(\Delta\phi) ; \dot{Y} \leftarrow XT \times \text{SIGN}(\Delta\psi) \times \text{SIGN}(\Delta\phi)$

ELSE

$\dot{X} \leftarrow YT \times \text{SIGN}(\Delta\psi) \times \text{SIGN}(\Delta\phi) ; \dot{Y} \leftarrow -XT \times \text{SIGN}(\Delta\psi) \times \text{SIGN}(\Delta\phi)$

ELSE

$\dot{\theta} \leftarrow -\dot{\phi} \times Z_2 \times \text{SIN}(\phi - \psi) / (Z_3 \times \text{SIN}(DIF))$

$\dot{X} \leftarrow -Z_2 \times \dot{\phi} \times \text{SIN}(\phi) - YT \times \dot{\theta} ; \dot{Y} \leftarrow Z_2 \times \dot{\phi} \times \text{COS}(\phi) + XT \times \dot{\theta}$

(the variable DM and DN are global and represent the rotated coordinates z_7 and z_8)

$X \leftarrow DM + Z_2 \times \text{COS}(\phi) + XT ; Y \leftarrow DN + Z_2 \times \text{SIN}(\phi) + YT$

$p \leftarrow (\dot{X} \times (XM - X) + \dot{Y} \times (YM - Y)) / \text{SQRT}(\dot{X}^2 + \dot{Y}^2)$

IF ABS(p) < @-8 THEN GO TO L4

(the next statement will prevent extended cycling)

IF TIME(2) - TM > 100 THEN

WRITE(message)

GO TO L4

IF $p \neq 0$ THEN

$BO3 \leftarrow \text{ABS}(DIF) < .15 \text{ OR } \text{ABS}(\text{ABS}(DIF) - \pi) < .15$

IF BO3 THEN

(control is given to the follower)

$\Delta\psi \leftarrow \text{ABS}(\Delta\psi) \times p \times \text{SIGN}(\dot{\phi}) / \text{ABS}(p \neq 0)$

IF $\phi' < \pi$ THEN

IF $\theta > \phi'$ EQV ((IF ABS(DIF) < @-9 THEN $\psi^* > \theta$ ELSE $\psi > \theta$) OR
(IF ABS(ABS(DIF) - π) < @-9 THEN $\psi^* + \pi < \theta$ ELSE $\psi + \pi > \theta$)) THEN

$\Delta\psi \leftarrow -\Delta\psi$

ELSE

IF $\theta < \phi'$ EQV ((IF ABS(DIF) < @-9 THEN $\psi^* < \theta$ ELSE $\psi < \theta$) OR
(IF ABS(ABS(DIF) - π) < @-9 THEN $\psi^* - \pi > \theta$ ELSE $\psi - \pi > \theta$)) THEN

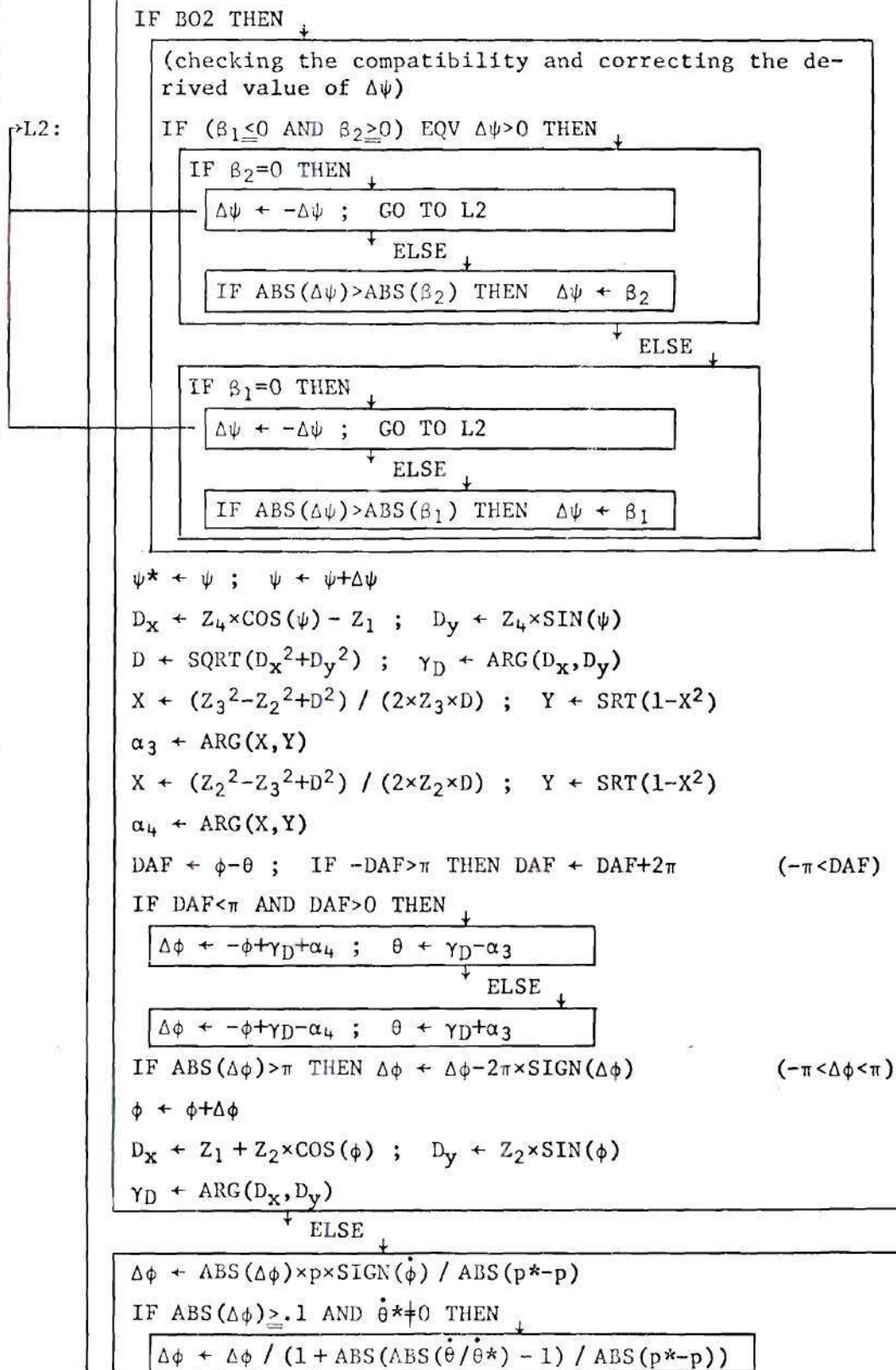
$\Delta\psi \leftarrow -\Delta\psi$

IF ABS($\Delta\psi$) > .5 THEN

$\Delta\psi \leftarrow .35 \times \text{SIGN}(\Delta\psi) / \text{SQRT}(K)$

(-0.35 < $\Delta\psi$ < 0.35)

SWIKL($Z_1, Z_4, Z_3, Z_2, \psi - \pi, BO2, \beta_1, \beta_2$)



IF $i=1$ OR $i=NU$ THEN ↓

$\Delta\phi \leftarrow \Delta\phi \times K / (K + .2)$

(statement required for the cusp
end of the Joukowski profile)

IF $ABS(\Delta\phi) > .5$ THEN ↓

$\Delta\phi \leftarrow .5 \times SIGN(\Delta\phi) / SQRT(K)$

$(-0.5 < \Delta\phi < 0.5)$

IF $BO1$ AND NOT $(p \neq 0$ AND $BO3)$ THEN ↓

(checking the compatibility and correcting the derived
value of $\Delta\phi$)

→L3: IF $(\alpha_1 \leq 0$ AND $\alpha_2 \geq 0)$ EQV $\Delta\phi > 0$ THEN ↓

IF $\alpha_2 = 0$ THEN ↓

$\Delta\phi \leftarrow -\Delta\phi$; GO TO L3

ELSE ↓

IF $ABS(\Delta\phi) > ABS(\alpha_2)$ THEN $\Delta\phi \leftarrow \alpha_2$

ELSE ↓

IF $\alpha_1 = 0$ THEN ↓

$\Delta\phi \leftarrow -\Delta\phi$; GO TO L3

ELSE ↓

IF $ABS(\Delta\phi) > ABS(\alpha_1)$ THEN $\Delta\phi \leftarrow \alpha_1$

IF $p \neq 0$ OR NOT $BO3$ THEN ↓

$D_y^* \leftarrow D_y$

$\phi \leftarrow \phi + \Delta\phi$; $\psi^* \leftarrow \psi$

$D_x \leftarrow Z_1 + Z_2 \times \cos(\phi)$; $D_y \leftarrow Z_2 \times \sin(\phi)$

$D \leftarrow SQRT(D_x^2 + D_y^2)$; $\gamma_D \leftarrow ARG(D_x, D_y)$

$X \leftarrow (Z_4^2 - Z_3^2 - D^2) / (2 \times Z_3 \times D)$; $Y \leftarrow SRT(1 - X^2)$

$\alpha_3 \leftarrow ARG(X, Y)$

$X \leftarrow (Z_3^2 - Z_4^2 - D^2) / (2 \times Z_4 \times D)$; $Y \leftarrow SRT(1 - X^2)$

$\alpha_4 \leftarrow ARG(X, Y)$

IF $D_y^* \geq 0$ THEN ↓

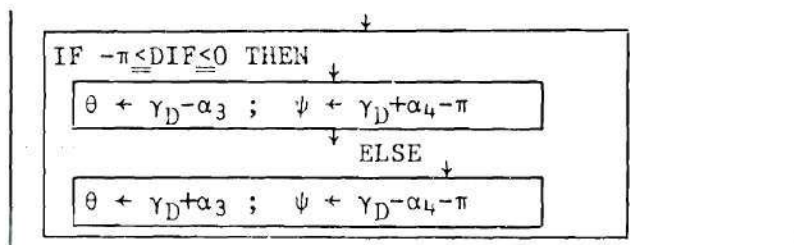
IF $\pi \geq DIF \geq 0$ THEN ↓

$\theta \leftarrow \gamma_D + \alpha_3$; $\psi \leftarrow \gamma_D - \alpha_4 + \pi$

ELSE ↓

$\theta \leftarrow \gamma_D - \alpha_3$; $\psi \leftarrow \gamma_D + \alpha_4 + \pi$

ELSE ↓



(an extra routine keeps the values in between $0 < \theta < 2\pi$ and $0 < \psi < 2\pi$)

$DIF \leftarrow \theta - \psi$; $\Delta\psi \leftarrow \psi - \psi^*$ ($\Delta\psi$ is kept within $-\pi < \Delta\psi < \pi$)

IF NOT ($ABS(DIF) < @-9$ OR $ABS(ABS(DIF) - \pi) < @-9$) THEN

IF $SIG \times SIGN(DIF) < 0$ THEN $\dot{\phi} \leftarrow -\dot{\phi}$
 $SIG \leftarrow SIGN(DIF)$

IF B01 THEN

$\alpha_2 \leftarrow \delta_2 - \phi$; $\alpha_1 \leftarrow \delta_1 - \phi$

$K \leftarrow K+1$

GO TO L1

→L4: (routine to establish the size and sign of ϵ)

$\epsilon \leftarrow \text{SQRT}((XM-X)^2 + (YM-Y)^2)$

IF ($ABS(\dot{X}) > @-9$ AND (($\dot{X} < 0$ AND $(YM-y) > 0$) OR ($\dot{X} > 0$ AND $(YM-Y) < 0$)))
 OR ($ABS(\dot{Y}) > @-9$ AND (($\dot{Y} < 0$ AND $(XM-X) < 0$) OR ($\dot{Y} > 0$ AND $(XM-X) > 0$)))
 THEN $\epsilon \leftarrow -\epsilon$

$\phi_1 \leftarrow \phi$; $\psi_1 \leftarrow \psi$

Computation of Gradient and Relaxation

The procedure for the calculation of the gradient is doing both, it determines the gradient and also a relaxation vector, whichever is of better use for the minimization process. The array ZA contains the required mechanism information, the array RA has the coordinates of the desired precision points. The output is contained in GRA the gradient vector, the real number SOM is its length and ORR is a listing of the errors between the precision points and the actual coupler point position. But the two last variables are also used as inputs, as it is required by the main program.

PROCEDURE GRAD(ZA,RA, GRA,SOM,ORR)

S ← 0 ; CO ← COR ← COS(Z₉) ; SI ← SIR ← SIN(Z₉)

CT ← IF SOM=0 THEN 0 ELSE 1

Q ← 9

FOR i←1 STEP 1 UNTIL 9 DO KC[i] ← i

→L1: FOR k←1 STEP 1 UNTIL (IF CT=0 THEN 1 ELSE Q) DO

IF CT≠0 THEN

KS ← KC[k]

SAM ← ZA[KS]

ZA[KS] ← ZA[KS] + (IF CT=1 THEN -DEN[KS]/2 ELSE DEN[KS]/2)

(DEN is global and holds the differentiation increments)

IF KS=9 THEN

CO ← COS(Z₉) ; SI ← SIN(Z₉)

ELSE

KS ← 1 ; SAM ← ZA[1]

φ ← 1

φ ← φ + .25 (for a proper error reading at the cusp of the Joukowski profile)

SWIKL(Z₁,Z₂,Z₃,Z₄,φ,BOL,α₁,α₂)

DM ← Z₇×CO + Z₈×SI + Z₁ ; DN ← -Z₇×SI + Z₈×CO

FOR i+1 STEP 1 UNTIL NU DO

$\Delta\phi \leftarrow$ IF i=1 THEN -.05 ELSE (IF CT=0 THEN $.5 \times \dot{\phi}$ ELSE
 $\text{ABS}(\text{PHI}[i] - \text{PHI}[i-1]) \times (\text{IF } i = \text{NU} \text{ THEN } .5 \text{ ELSE } 1) \times \text{SIGN}(\dot{\phi}))$
 $\text{ERROR}(\text{RA}[1,i], \text{RA}[2,i], \text{ZA}, \alpha_1, \alpha_2, \text{BO1}, i, \text{ER}[CT, \text{KS}, i], \text{PHI}[i], \text{PSI}[i])$

$\phi \leftarrow \text{PHI}[1]$; $\psi \leftarrow \text{PSI}[1]$

$\text{ZA}[\text{KS}] \leftarrow \text{SAM}$

CO \leftarrow COR ; SI \leftarrow SIR

CT \leftarrow CT+1

IF CT \leq 2 THEN GO TO L1

IF SOM=0 THEN

FOR i+1 STEP 1 UNTIL NU DO $\text{ORR}[i] \leftarrow \text{ER}[0,1,i]$

FOR i+1 STEP 1 UNTIL NU DO

$\text{ERR}[0,1,i] \leftarrow \text{ABS}(\text{ORR}[i])$

CT \leftarrow Q ; Q \leftarrow 0 (KS and CT have now a new meaning)

IF S \geq 3 THEN (required for cusp of Joukowski profile)

K \leftarrow 2 ; KS \leftarrow NU-1

ELSE

K \leftarrow 1 ; KS \leftarrow NU

FOR j+1 STEP 1 UNTIL 9 DO (checking the increment size)

SAM \leftarrow DEN[j] ; BO2 \leftarrow BO3 \leftarrow FALSE

FOR i+K STEP 1 UNTIL KS DO

$\text{DER}[j,i] \leftarrow (\text{ER}[2,j,i] - \text{ER}[1,j,i]) / \text{SAM} \times \text{SIGN}(\text{ORR}[i])$

(the sign of $\text{DER}[j,i] = \partial \epsilon_i / \partial z_j$ will indicate whether or not the increment $\text{DEN}[j] = \Delta z_j$ represents an improvement)

$\text{ERR}[j,i] \leftarrow \text{SAM}^2 / 2 + 2 \times (\text{ORR}[i] - \text{ER}[1,j,i]) \times (\text{ER}[2,j,i] - \text{ORR}[i])$

IF $\text{ERR}[j,i] < 0$ THEN

BO2 \leftarrow TRUE

ELSE

IF $\text{ERR}[j,i] < \text{SAM} / .15 \times \text{ABS}(\text{ER}[2,j,i] - 2 \times \text{ORR}[i] + \text{ER}[1,j,i])$
 THEN BO3 \leftarrow TRUE (see equation 81)

$\text{ERR}[j,i] \leftarrow \text{ERR}[j,i] / (\text{SAM}^2 / 2 + (\text{ORR}[i] - \text{ER}[1,j,i])^2$
 $+ (\text{ER}[2,j,i] - \text{ORR}[i])^2)$

IF BO2 OR BO3 THEN

Q ← Q+1 ; KC[Q] ← j

DEN[j] ← IF BO2 THEN SAM/4 ELSE SAM/2

S ← S+1 (count for differentiation attempts)

IF Q>0 THEN

CT ← IF S=3 THEN (SOM←0) ELSE 1

GO TO L1

FOR j+1 STEP 1 UNTIL 9 DO

FOR i+1 STEP 1 UNTIL NU DO

DER[j,i] ← DER[j,i]×ERR[j,i]

(DER[j,i] = $\partial \epsilon_i / \partial z_j$)

LSQ(DER,NU,9,ER[0,1,*],GRA)

SOM ← 0

FOR i+1 STEP 1 UNTIL 9 DO

SOM ← SOM + GRA[i]²

SOM ← SQRT(SOM)

(length of gradient vector)

IF SOM<15 THEN

BU ← FALSE

(BU is a global Boolean)

ELSE

IF SOM* < SOM THEN BU ← NOT BU

(SOM* is a global variable)

IF BU THEN

(begin of relaxation method)

SAM ← ER[0,1,1]

(establishing the largest error ϵ)

FOR i+2 STEP 1 UNTIL NU DO

IF SAM < ER[0,1,i] THEN

SAM ← ER[0,1,i]

SAM ← SAM×KT/(KT+10)

(KT is a global count of the iterations)

KS ← 0

FOR i+1 STEP 1 UNTIL NU DO

(find the group of larger errors for the relaxation)

IF SAM < ER[0,1,i] THEN

KS ← KS+1

QC[KS] ← i

FOR j+1 STEP 1 UNTIL 9 DO

FOR i+1 STEP 1 UNTIL NU DO

ERR[j,i] ← DER[j,i]×ER[0,1,i]

```

    SAM ← SUM ← 0
    FOR i←1 STEP 1 UNTIL KS DO
        SAM ← SAM + ERR[j, QC[i]]
        SUM ← SUM + DER[j, QC[i]]2
    GRA[j] ← SAM/SUM
    (relaxation vector)

SOM* ← SOM
SOM ← 0
FOR i←1 STEP 1 UNTIL 9 DO
    SOM ← SOM + GRA[i]2
SOM ← SQRT(SOM)
    (length of relaxation vector)
IF SOM > 1 THEN
    (reduction to a unit vector)
    FOR i←1 STEP 1 UNTIL 9 DO
        GRA[i] ← GRA[i]/SOM
    SOM ← 1
IF S > 4 THEN
    (regenerating the increment size Δzi)
    FOR i←1 STEP 1 UNTIL CT DO
        DEN[KC[i]] ← DEN[KC[i]] × 2*(S-1)

```


Main Program

The main program needs a certain number of read-in statements which were given by data cards but also by a few constant values and statements in the program itself which have to be modified to suit the attempted synthesis problem. Explanatory notes are given in parenthesis at the appropriate locations.

READ (NU) (NU is the number of desired precision points)

READ (FOR j+1 STEP 1 UNTIL NU DO FOR i+1,2 DO RA[i,j])

(RA contains the coordinates of all desired precision points)

JOUK(a,s,m,NU,RA)

FOR i+1 STEP 1 UNTIL NU DO RA[1,i] ← RA[1,i] + Δx

(an alternate method to calculate the precision point coordinates, here for the Joukowski profile with a suitable transformation for proper positioning)

READ (FOR i+1 STEP 1 UNTIL 9 DO ZA[i])

(ZA contains the physical dimensions of the initial mechanism as shown in Fig. 12, but this read statement could be replaced by an initial four-accuracy point procedure with its subroutines as shown earlier)

READ (φ,ψ)

(these angles will indicate the initial position of the mechanism and have to be given or calculated from other datas)

READ (FOR i+1 STEP 1 UNTIL 9 DO DEN[i])

(reads datas for increments suitable for the differentiation)

SOM* ← SOM ← 0 ; KT ← KO ← 0

BU ← FALSE

→B1: KT ← KT+1 (count for number of iterations)

GRAD(ZA,RA,RES,SOM,ER[0,*])

← IF SOM < 2@-3 THEN GO TO B4 (the small size of the gradient vector indicates the end of the minimization)

d* ← d + h + Δh* ← Δh ← 0

KN ← 0

BD1 ← 0 ; BD2 ← 8 (BD1 and BD2 are boundaries for h)

∂h ← SOM × .03

FOR i+1 STEP 1 UNTIL 9 DO ZA[i] ← ZA[i]

→B2: S ← CT ← 0

d** ← d* ; d* ← d

→B3: IF CT≠0 OR d≠0 THEN ↓

IF CT=0 THEN ↓

FOR i←1 STEP 1 UNTIL 9 DO ZO[i] ← ZA[i]

ELSE ↓

FOR i←1 STEP 1 UNTIL 9 DO ↓

ZO[i] ← ZA[i] - (IF CT=1 THEN -RES[i]×∂h/2 ELSE RES[i]×∂h/2)

φ̇ ← 1

φ ← φ+.25 (needed for Joukowski profile)

SWIKL(ZO₁,ZO₂,ZO₃,ZO₄,φ,BO,α₁,α₂)

DM ← ZO₇×COS(Z₉) + ZO₈×SIN(Z₉) + ZO₁

DN ← -ZO₇×SIN(Z₉) + ZO₈×COS(Z₉)

FOR i←1 STEP 1 UNTIL NU DO ↓

Δφ ← IF i=1 THEN -.05 ELSE ABS(PHI[i]-PHI[i-1])

× (IF i=NU THEN .5 ELSE 1) × SIGN(φ̇)

ERROR(RA[1,i],RA[2,i],ZO,α₁,α₂,BO,i,ER[CT,i],PHI[i],PSI[i])

φ ← PHI[1] ; ψ ← PSI[1]

CT ← CT+1

←— IF CT≤2 THEN GO TO B3

DAM←DUM←DOM ← 0 (start of differentiation method)

FOR i←1 STEP 1 UNTIL NU DO ↓

DAM ← DAM + ER[1,i]²

DUM ← DUM + ER[2,i]²

DOM ← DOM + ER[0,i]²

DAM ← SQRT(DAM) ; DUM ← SQRT(DUM) ; DOM ← SQRT(DOM)

(this should help to increase the slope towards the desired min.)

DER ← (DUM-DAM)/∂h

BO ← FALSE

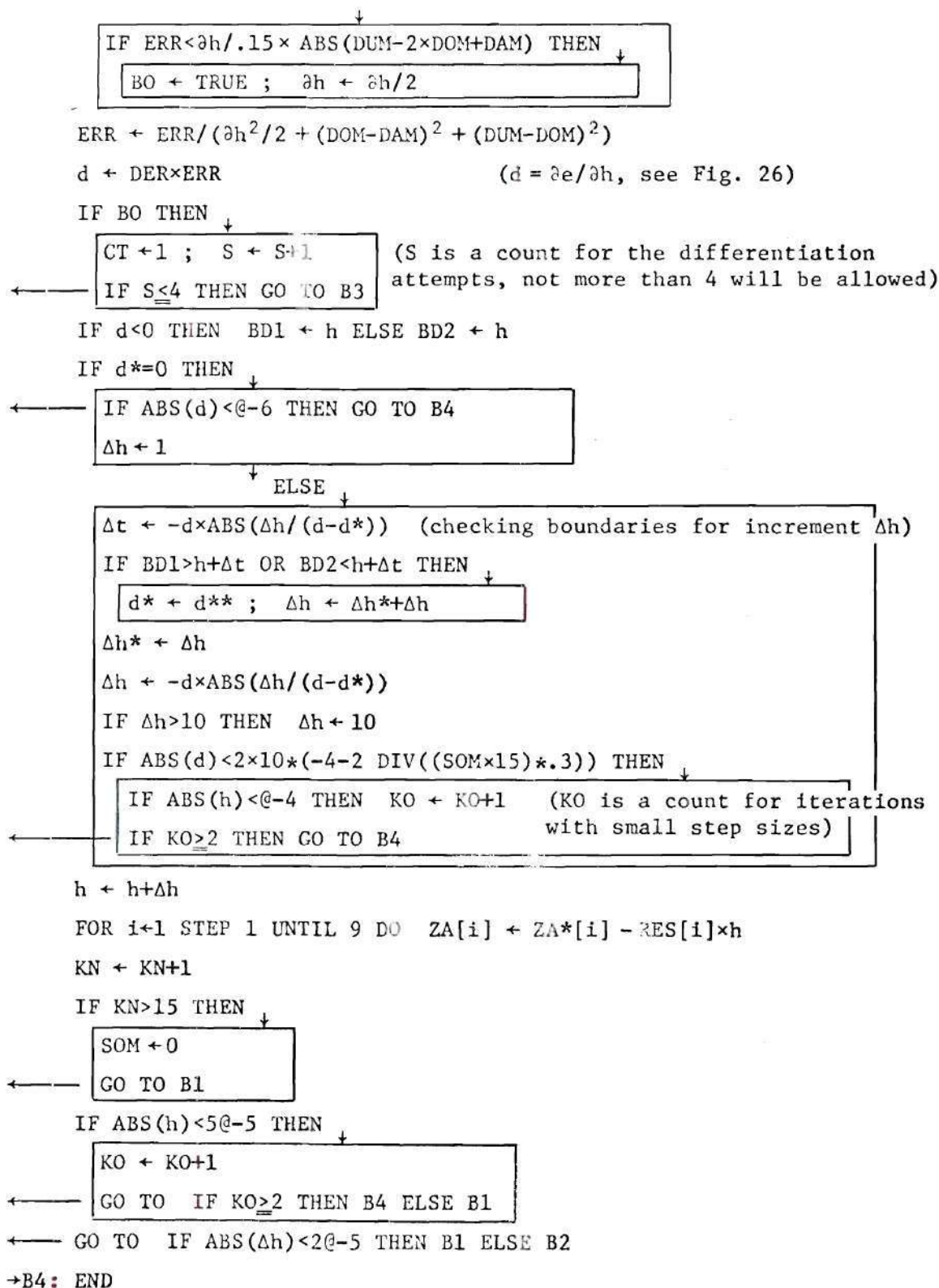
ERR ← ∂h²/2 + 2×(DOM-DAM)×(DUM-DOM)

IF ERR<0 THEN ↓

BO ← TRUE

∂h ← h/4

ELSE ↓



CHAPTER XI

CONCLUSIONS

It was found that the method related to the steepest descent or gradient method is not powerful enough to warrant a solution to any given problem. Reasons for this behavior are quoted and explained. It is trusted that the addition of the relaxation method represents a more powerful tool in optimization attempts and its behavior can be explained by its tendency to take into consideration only major trends or characteristics of the given problem, even when the problem has a multitude of variables and is of a highly nonlinear character.

The material presented shows that a hybrid computer combination would have been a very effective means of solving the synthesis problem. The limitations of equipment required other methods to be developed which can never achieve as fast an operational speed. But the basic ideas here presented should be transferable later to hybrid use without too many difficulties, once the required hardware becomes available.

It was interesting to observe that, although a given set of nine points should specify uniquely a plane four-bar mechanism with its additional two cognate solutions, it is difficult to establish the proper sequence of the given nine points because of the various forms the coupler curve can assume. This accounts for the failure to obtain a perfect solution for any given set of nine accuracy points. Generally, it should be possible to apply an analog system for the solution of nine

variables given by nine equations. This method, however, would restrict itself to a maximum of only nine points and because of this limitation was not considered in this investigation.

If there exists an obtainable maximum of only nine accuracy points, then it is noted that there are only eight intersection points between the desired and obtained curve in the presented airfoil problem. Every pair of closed curves can have only an even number of intersections, but here the limit is nine, hence only eight can occur. A number smaller than eight would indicate a poor approximation and so could be used as a measure for the quality of an obtained result should the computer program be interrupted before its final steps.

We turn now to consider a possible future development which may lead to a better simulator. This germ of an idea derives from consideration of the tricircular nature of every four-bar coupler curve, and the three cognate linkages for every curve. An analogy is drawn to the motion of a point following three circles which correspond to the absolute rotation of the driver, the follower, and the relative rotation of the coupler. This idea might lead to the discovery of a suitable transformation technique to enable the coupler link motion to be expressed as an absolute motion and so strip the computations of a large number of logic and numerical statements. Such a discovery would increase the achieved speed and simplify considerably the entire computations.

LIBLIOGRAPHY

Literature Cited

1. Peaucellier, M.
"Correspondance (Compas Composé)"
Nouvelles Annales 2^{ième} ser., vol. 3, pp. 414-16 (1864).
2. Peaucellier, M.
"Note sur une question de Géométrie de Compas"
Nouvelles Annales 2^{ième} ser., vol. 12, pp. 71-78 (1873).
3. Lavery, W. H.
"Extension of Peaucellier's Theorem"
Proceedings of L.M.S., vol. VI, pp. 84-85 (Feb. 1875).
4. Tesar, Delbert
"The Analytical Theory of Coplanar Motion Applied to Approximate Four-Bar Straight Line Mechanisms"
Doctoral Dissertation, School of Mechanical Engineering, Georgia Institute of Technology (June 1964).
5. Alt, H.
"Zur Synthese der ebenen Mechanismen"
Zeitschrift für angewandte Mathematik und Mechanik, vol. 1, pp. 373-98 (Oct. 1921).
6. Alt, H.
"Über die Erzeugung gegebener ebener Kurven mit Hilfe des Gelenk-viereckes"
Zeitschrift f. Math. und Physik, Band 3, pp. 13-18 (1923).
7. Cayley, A.
"On the Mechanical Description of a Nodal Bicircular Quartic"
Proceedings of L.M.S., vol. III, pp. 100-06 (May 1870).
8. Cayley, A.
"On the Mechanical Description of Certain Sextic Curves"
Proceedings of L.M.S., vol. IV, pp. 105-11 (April 1872).
9. Cayley, A.
"On Three-Bar Motion"
Proceedings of L.M.S., vol. VII, pp. 136-66 (March 1876).
10. Roberts, Samuel
"On Three-Bar Motion in Plane Space"
Proceedings of L.M.S., vol. VII, pp. 14-23 (Nov. 1875).

BIBLIOGRAPHY (Continued)

11. Reuleaux, F.
Kinematics of Machinery, a translation of the German text
Theoretische Kinematik
 MacMillan and Co., London, 1876.
12. Burmester, Ludwig Ernst Hans
Lehrbuch der Kinematik, erster Band (Die ebene Bewegung)
 Verlag von Arthur Felix, Leipzig, 1888.
13. Grübler, M.
"Die Krümmungsradien der Polbahnen"
Zeitschrift f. Math. und Physik, Band 34, pp. 305-10 and Band 29,
 pp. 212 and 382 (1889).
14. Grübler, M.
"Ueber die Kreispunkte einer coplan bewegten Ebene"
Zeitschrift f. Math. und Physik, Band 37, pp. 35-56 (1892).
15. Müller, Reinhold
"Ueber die Doppelpunkte der Koppelkurve"
Zeitschrift f. Math. und Physik, Band 34, pp. 303-05 and pp. 372-75;
 Band 36, pp. 65-70 (1889).
16. Müller, Reinhold
"Ueber die Krümmungsmittelpunkte der Bahnkurven in ebenen ähnlich-
 veränderlichen Systemen"
Zeitschrift f. Math. und Physik, Band 36, pp. 129-37 (1891).
"Ueber die Krümmung der Bahnevoluten der starren ebenen Systemen"
Zeitschrift f. Math. und Physik, Band 36, pp. 193-205 (1891).
"Konstruktion der Krümmungsmittelpunkte der Hüllbahnevoluten bei
 starren ebenen Systemen"
Zeitschrift f. Math. und Physik, Band 36, pp. 257-66 (1891).
17. Müller, Reinhold
"Ueber die Bewegung eines starren ebenen Systems durch 5 unendlich
 benachbarten Lagen"
Zeitschrift f. Math. und Physik, Band 37, pp. 129-50 (1892).
18. Müller, Reinhold
Einführung in die theoretische Kinematik
 Springer Verlag, Berlin, 1932.
19. Darboux, G.
"De l'emploi des Fonctions Elliptiques dans la Théorie du Quadri-
 latère Plan"
Bulletin des Sciences Mathématiques, vol. 3, pp. 109-36 (1879).

BIBLIOGRAPHY (Continued)

20. Morley, F. V.
"An Analytical Treatment of the 3-bar Curve"
Proceedings of L.M.S., vol. 21, pp. 140-60 (June 1921).
21. Bereis, R.
"Aufbau einer Theorie der ebenen Bewegung mit Verwendung komplexer Zahlen"
Oesterreichische Ingenieur Archiv, vol. 5, pp. 246-66 (March 1951).
22. Rosenauer, N.
"Anwendung von komplexen Veränderlichen zur Synthese einer Kurbelschwinge mit vorgeschriebenen Grenzen der Abtriebs-Winkel-Geschwindigkeit"
Ingenieur-Archiv, vol. 24, pp. 43-46 (June 1955).
23. Sieker, K. H.
"Analytische Betrachtung des Gelenkviereckes insbesondere der Burmesterschen Punkte"
VDI Berichte, Band 5, pp. 55-60 (1955).
24. Sieker, K. H.
"Zur algebraischen Mass-Synthese ebener Kurbelgetriebe"
Ingenieur-Archiv, Band 24, 1. Teil, pp. 188-215; 2. Teil, pp. 233-57 (1956).
25. Bennett, G. T.
"The Three-Bar Sextic Curve"
Proceedings of L.M.S., vol. 20, ser. 2, pp. 59-84 (April 1920).
26. Beyer, Rudolf
Technische Kinematik
Verlag von Johann Ambrosius Barth, Leipzig, 1931, published by J. W. Edwards, Ann Arbor, Michigan, 1948.
27. Beyer, Rudolf
Kinematische Getriebesynthese (Grundlagen einer quantitativen Getriebelehre ebener Getriebe)
Springer-Verlag, Berlin, 1953.
28. Chebichev, P. L.
Oeuvres de P. L. Tchebyschew
Les soins de Mrs. A. Markoff et Sonin, vol. 2, St. Pétersbourg, 1907 (1899).
29. Bloch, S. Sch.
Angenäherte Synthese von Mechanismen
Verlag Technik, Berlin, 1951.

BIBLIOGRAPHY (Continued)

30. Meyer zur Capellen, W.
 "Der Flächeninhalt von Koppelkurven, ein Beitrag zu ihrem Formenwandel"
Forschungsberichte des Wirtschafts- und Verkehrsministeriums, Nr. 506,
 Westdeutscher Verlag, Köln, 1958.
31. Meyer zur Capellen, W.
 "Die zweidimensionale Fourieranalyse spezieller Koppelkurven"
Zeitschrift f. Math. und Mech., vol. 39, pp. 31-40 (Feb. 1958).
32. Meyer zur Capellen, W.
 "Harmonische Analyse an Kurbeltrieben"
Konstruktion, Band 12, Heft 1, pp. 38-41 (1960).
33. Meyer zur Capellen, W.
 "Calculation- and Instrument-Procedures for the Harmonic Analysis
 of Periodic Mechanisms"
 Proceedings of Intern. Conf. for Teachers of Mechanisms, Yale Uni-
 versity, pp. 13-42 (1961).
34. Hain, Kurt
 "Punktlagenreduktion als getriebesynthetisches Hilfsmittel"
Maschinenbau/Betrieb, Beilage Getriebetechnik 11, Heft 1, pp. 29-
 31 (1943).
35. Artobolevskii, I. I., Blokh, S. Sh., and Dobrovolskii, V. V.
Synthesis of Mechanisms
 State Publishing House, Moscow and Leningrad USSR, pp. 253-374, 1944.
36. Hirschhorn, Jeremy
Kinematics and Dynamics of Plane Mechanisms
 McGraw-Hill Book Co., New York, 1962.
37. Sandor, George N.
 "A General Complex-Number Method for Plane Kinematic Synthesis with
 Applications"
 Dissertation, Columbia University, Engineering Science (1959).
38. Freudenstein, Ferdinand and Sandor, George N.
 "Synthesis of Path-Generating Mechanisms by Means of a Programmed
 Digital Computer"
 ASME Trans., ser. B, vol. 81, no. 2, pp. 159-68 (May 1959).
39. Freudenstein, Ferdinand
 "Structural Error Analysis in Plane Kinematic Synthesis"
 ASME Trans., ser. B, vol. 81, pp. 15-22 (Feb. 1959).
40. Fox, R. L. and Willmert, K. D.
 "Optimum Design of Curve-Generating Linkages with Inequality Con-
 straints"
 ASME Paper no. 66-Mech-20 (Nov. 1966).

BIBLIOGRAPHY (Continued)

41. McLarnan, C. W.
"Synthesis of Six-Link Plane Mechanisms by Numerical Analysis"
ASME Paper no. 61-WA-86 (1961).
42. Kiss, Istvan S.
"Computer-Aided Kinematic Synthesis of a General Planar Mechanism"
ASME Paper no. 66-Mech-40 (1966).
43. Timko, Charles A.
"Determining Linkage Proportions"
Machine Design, pp. 127-30 (March 1966).
44. Han, Chi-Yeh
"A General Method for the Optimum Design of Mechanisms"
Journal of Mechanisms, vol. 1, Pergamon Press (1966).
45. Suh, C. H. and Radcliffe, C. W.
"Synthesis of Plane Linkages with Use of the Displacement Matrix"
ASME Paper no. 66-Mech-19 (Nov. 1966).
46. Timm, R. F.
"Analog Simulation of Rigid Link Mechanisms"
ASME Paper no. 66-Mech-19 (Nov. 1966).
47. Crossley, F. Erskine
"Geometric Computing: Analogue-Simulation of a Linkage"
International Journal of Mech. Science, vol. 7, no. 9, pp. 595-601
(April 1964).
48. Crossley, F. Erskine
"Die Nachbildung eines mechanischen Kurbelgetriebes mittels eines elektronischen Analogrechners"
Feinwerktechnik, 67 (June 1963).
49. Lenk, E. W.
"Instrumentelle und elektrische Verfahren zur Erzeugung und Aufzeichnung von Koppelkurven"
Konstruktion, Heft 10, pp. 393-96 (Nov. 1962).
50. Keller, E. Robert
"Mechanism Design by Electronic Analog Computer"
Trans. 7th Conference on Mechanisms, Purdue University, Penton Publ. Co., Cleveland, Ohio, pp. 11-21 (Oct. 1962).
51. Freudenstein, Ferdinand and Sandor, George N.
"On the Burmester Points of a Plane"
ASME Trans., ser. E, vol. 83, pp. 41-49 (March 1961).

BIBLIOGRAPHY (Continued)

52. Roberts, Samuel
 "On the Mechanical Description of Some Species of Circular Curves
 of the 3rd and 4th Degrees"
 Proceedings of L.M.S., vol. II, pp. 125-36 (Jan. 1869).
53. Roberts, Samuel
 "On the Motion of a Plane Under Certain Conditions"
 Proceedings of L.M.S., vol. III, pp. 286-318 (June 1871).
54. Laguerre, E. N.
Sur les Courbes Planes Algébriques
 Oeuvres de Laguerre vol. II, Gauthier-Villars, Paris, 1905.
55. Hrones, J. A. and Nelson, G. L.
Analysis of the Four-Bar Linkage (Atlas of Coupler Curves)
 John Wiley & Sons, Inc., New York, 1951.
56. Grashof, F.
Theoretische Maschinenlehre
 Berlin 1883, p. 117, 1883.
57. Stiefel, Edward L., Rutishauser, H., Ginsburg, Th., and Engeli, M.
 "Refined Iterative Methods for Computation of the Solution and the
 Eigenvalues of Self-Adjoint Boundary Value Problems"
 Mitteilungen aus dem Institut für angewandte Mathematik an der ETH
 Zürich, Birkhäuser Verlag, Basel, 1959.
58. Southwell, R. V.
Relaxation Methods in Engineering Science
 Oxford University Press, London, 1940.
59. Southwell, R. V.
Relaxation Methods in Theoretical Physics
 Oxford University Press, London, 1946.
60. Linebarger, R. N.
 "Digital Simulation Techniques for Direct Digital Control Studies"
 19th Annual ISA Conference and Exhibit, New York, Preprint no. 5.4-
 3-64 (Oct. 1964).
61. Vichnevetsky, R. and Waha, J. P.
 "Dynamic Programming and Hybrid Computation"
Simulation, pp. 34-43 (Nov. 1964).
62. Linebarger, Robert N. and Brennan, Robert D.
 "A Survey of Digital Simulation: Digital Analog Simulator Programs"
Simulation, 3, no. 6, pp. 22-35 (1964).

BIBLIOGRAPHY (Continued)

63. Brennan, Robert D.
"An Evaluation of Digital Analog Simulator Languages"
IFIP Congress 1965, New York, Proceedings vol. 27, part II (May 1965)
64. Linebarger, Robert N. and Brennan, Robert D.
"Digital Simulation for Control System Design"
Proceedings of the 6th Joint Automatic Control Conference, Troy,
N. Y. (June 1965).
65. Teichroew, Daniel and Lubin, Francis John
"Computer Simulation - Discussion of the Technique and Comparison
of Languages"
Communications of the ACM, vol. 9, no. 10, pp. 723-39 (Oct. 1966).
66. Lewis, D. W. and Gyory, C. K.
"Kinematic Synthesis of Plane Curves"
ASME Paper no. 66-Mech-13 (Nov. 1966).
67. Greville, T. N. E.
"The Pseudoinverse of a Rectangular or Singular Matrix and Its
Application to the Solution of Systems of Linear Equations"
SIAM Review, vol. 1, no. 1, pp. 38-43 (Jan. 1959).
68. Diederich, Franklin W.
"A Least-Squares Solution of a Linear System"
SIAM Review, vol. 6, pp. 182-83 (1964).
69. Zurmühl, Rudolf
Matrizen und ihre technische Anwendung
Springer-Verlag, Berlin, 4th edition, 1964.
70. Penrose, R.
"A Generalized Inverse for Matrices"
Proceedings of Cambridge Philosophical Society, vol. 51, pp. 406-
13 (1955).
71. Tomáš, Josef
"The Synthesis of Mechanism as a Nonlinear Programming Problem"
Journal of Mechanisms, vol. III, no. 3, Autumn 1968, to be published.
72. Garrett, Richard E.
"Optimal Synthesis of Randomly Generated Linkages"
Trans. of ASME, Journal of Engineering for Industry, Paper no.
67-WA/DE-10 (1967).
73. Künzi, Hans Paul and Krelle, Wilhelm
Nichtlineare Programmierung
Springer-Verlag, Berlin, 1962.

BIBLIOGRAPHY (Continued)

74. Stiefel, Eduard L.
An Introduction to Numerical Mathematics
Academic Press, London, New York, 1963.
75. Mesch, F.
"Als Gast in Kiew (Kybernetik)"
VDI Nachrichten, Jahrgang 22/14, p. 1 (April 1968).

Other References

76. Allen, D. N. de G.
Relaxation Methods
McGraw-Hill Book Co., Inc., New York, 1954.
77. Beyer, Rudolf
Kinematische-getriebeanalytisches Praktikum
Springer-Verlag, Berlin, 1958.
78. Bodewig, E.
Matrix Calculus
North-Holland Publishing Co., Amsterdam, 1959.
79. Bricard, R.
Lecons de Cinématique
Gauthier-Villars, Vol. II, Paris, 1927.
80. Conte, S. D.
Elementary Numerical Analysis
McGraw-Hill Book Co., New York, 1965.
81. Faddeev, D. K. and Faddeeva, V. N.
Computational Methods of Linear Algebra
Translation by Robert C. Williams (Russian)
W. H. Freeman and Company, San Francisco and London, 1963.
82. Giloi, Wolfgang and Lauber, Rudolf
Analogrechnen
Springer-Verlag, Berlin, 1963.
83. Graybill, Franklin A.
An Introduction to Linear Statistical Models
McGraw-Hill Book Co., New York, 1961.
84. Hartee, D. R.
Numerical Analysis
At the Clarendon Press, Oxford, 1958.

BIBLIOGRAPHY (Continued)

85. Hartenberg, Richard S. and Denavit, Jacques
Kinematic Synthesis of Linkages
McGraw-Hill Book Co., New York, 1964.
86. Hilbert, D. and Cohn-Vosslen, S.
Geometry and the Imagination
Chelsea Publishing Co., New York, 1956.
87. Hilton, Harold
Plane Algebraic Curves
Oxford University Press, London, 1932.
88. Householder, Alston S.
Principles of Numerical Analysis
McGraw-Hill Book Co., New York, 1953.
89. Kraus, R.
Grundlagen der Getriebelehre
Wolfenbüttel, 1949.
90. Lavi, Abraham and Vogel, Thomas P.
Recent Advances in Optimization Techniques
John Wiley & Sons, New York, 1965.
91. Lenk, E.
Elektrisches Analogiegerät zur Fourier-Analyse und Gelenkgetrieben
Forschungsberichte des Landes Nordrhein-Westfalen, Köln: West-deutscher Verlag, 1960.
92. Levenberg, Kenneth
"A Method for the Solution of Certain Non-Linear Problems in Least Squares"
Quarterly of Applied Mathematics, vol. II, no. 2, pp. 164-68 (1944).
93. Levine, Leon
Methods for Solving Engineering Problems Using the Analog Computer
McGraw-Hill Book Co., New York, 1964.
94. Rogers, A. E. and Connolly, T. W.
Analog Computation in Engineering Design
McGraw-Hill Book Co., New York, 1960.
95. Roth, Bernard and Freudenstein, Ferdinand
"Synthesis of Path-Generating Mechanisms by Numerical Methods"
Trans. ASME, Journal of Engineering for Industry, pp. 298-306
(August 1963).

BIBLIOGRAPHY (Concluded)

96. Sadd, G. D.
"On Certain Types of Plane Unicursal Sextic Curves"
Proceedings of L.M.S., vol. 21, pp. 331-342 (Jan. 1922).
97. Salmon, George
Higher Plane Curves
G. E. Stechert & Co., New York, 1934.
98. Schönflies, A.
Einführung in die Analytische Geometrie der Ebene und des Raumes
Springer-Verlag, Berlin, 1931.
99. Shaw, F. S.
An Introduction to Relaxation Methods
Dover Publications, 1953.

VITA

August J. Nechi was born on the 10th October 1916 in Vienna, the capital of Austria. There he attended all schools required for an academic career. This education was terminated by obtaining the academic degree of a Diplom-Ingenieur in Mechanical Engineering (Maschinenbau) from the Institute of Technology in Vienna (Technische Hochschule in Wien) in May 1944, just about one year before the end of the big power struggle on the European continent.

He married Maria Schoderböck, a native of Austria, in September 1942 and emigrated with his spouse to Canada in 1951 and finally to the United States in 1961.

He has a long experience as practical engineer which has covered a wide scope of activities. He first held a position as research engineer with the Institute of Technology in Vienna. After the war he held the position of a maintenance mechanic in Switzerland but became soon a design engineer with a large machine factory in Winterthur, Switzerland.

In Canada he held jobs with various consulting firms for which he worked in various capacities from a design engineer up to a project engineer for mining plants. When coming to the United States he was appointed to a teaching position as Assistant Professor in the Mechanical Department of the Tuskegee Institute in Alabama. Feeling that he could be useful to the teaching profession because of his broad background in theoretical and practical work and trying to satisfy the status of this profes-

sion he entered the Graduate Division of the Georgia Institute of Technology in September 1964 to pursue the Doctor of Philosophy degree in Mechanical Engineering.

His primary interest is a career which would permit a combined use of his theoretical and practical background which starts from the manual work up to the supervisory position of a project engineer but which would also provide sufficient incentive to continue the research activities of his field in machine design.